On the Fundamental Limits of Cache-Aided Multiuser Private Information Retrieval

Xiang Zhang, *Student Member, IEEE*, Kai Wan, *Member, IEEE*, Hua Sun, *Member, IEEE*, Mingyue Ji, *Member, IEEE*, and Giuseppe Caire, *Fellow, IEEE*

***Abstract*— We consider the problem of cache-aided Multiuser Private Information Retrieval (MuPIR) which is an extension of the single-user cache-aided PIR problem to the case of multiple users. In cache-aided MuPIR, each of the K_{u} cache-equipped users wishes to privately retrieve a message out of K messages from N databases each having access to the entire message library. Demand privacy requires that any individual database learns nothing about the demands of all users. The users are connected to each database via an error-free shared-link. In this paper, we aim to characterize the optimal trade-off between user cache memory and communication load for such systems. First, we propose a novel approach of *cache-aided interference alignment (CIA)*, for the MuPIR problem with $K = 2$ messages, $K_{\mathrm{u}} = 2$ users and $N \geq 2$ databases. The CIA approach is optimal when the cache placement is uncoded. For general cache placement, the CIA approach is optimal when $N = 2$ and 3 verified by the computer-aided converse approach. Second, for the general case, we propose a *product design* (PD) which incorporates the PIR code into the linear caching code. The product design is shown to be order optimal within a multiplicative factor of 8 and is exactly optimal in the high memory regime.**

***Index Terms*— Private information retrieval, coded caching, interference alignment, multiuser.**

I. Introduction

Introduced by Chor *et al.* in [3], the problem of private information retrieval (PIR) seeks efficient ways for a user to retrieve a desired message from N databases, each holding a library of K messages, while keeping the desired message's identity private from each database. Sun and Jafar (SJ) recently characterized the capacity of the PIR problem with non-colluding databases [4], [5]. Coded caching was originally proposed by Maddah-Ali and Niesen (MAN) in [6] for a shared-link caching network consisting of a server, which is connected to K_{u} users through a noiseless broadcast channel and has access to a library of K equal-length files. Each user can cache M files and requests one file. The MAN scheme proposed a combinatorial cache placement design so that during the delivery phase, each transmitted coded message is simultaneously useful to multiple users such that the communication load can be significantly reduced. Under the constraint of uncoded cache placement and for worst-case load, the MAN scheme was proved to be optimal when the number of files is no less than the number of users [7] and order optimal within a factor of two in general [8].

The combination of privacy and caching, sometimes referred to as side information, has gained significant attentions recently. Two different privacy models are commonly considered. First, in [9]–[17], the *user-against-database* privacy model was studied where individual databases are prevented from learning the single user's demand. The author in [9] studied the case where a single cache-aided user is connected to a set of N replicated databases and showed that memory-sharing between the memory-load[1] pairs $(0, 1+\frac{1}{N}+\cdots+\frac{1}{N^{K-1}})$ and $(K, 0)$ (i.e., split the messages and cache memories proportionally and then implement two PIR schemes on the two independent parts of the messages) is actually optimal if the databases are aware of the users' cached content. However, if the databases are unaware of the user's cached content, then there is a multiplicative "unawareness gain" in capacity in terms of the user memory as shown in [10], [11]. Different from [9]–[11] where the cache can be arbitrary functions of the messages, the user's cache is restricted to be in the form of M full messages in [12]–[18]. Along this line, two models are usually studied, referred to as PIR-SI (PIR with non-private side information) where the cached messages are not known by the databases but the privacy of the cache needs not to be preserved, and PIR-PSI (PIR with private side information) where the joint privacy of both the desired message and the cache needs to be preserved. For PIR-SI with a single database ($N = 1$), the capacity is shown

Manuscript received October 13, 2020; revised April 21, 2021; accepted June 14, 2021. Date of publication June 22, 2021; date of current version September 16, 2021. The work of Xiang Zhang and Mingyue Ji was supported through the National Science Foundation grants CCF-1817154, SpecEES-1824558, and the Idaho National Laboratory (INL) Laboratory Directed Research and Development (LDRD) Program under DOE Idaho Operations Office Contract DE-AC07-05ID14517. The work of Kai Wan and Giuseppe Caire was supported in part by the European Research Council under the ERC Advanced Grant No. 789190, CARENET. The work of Hua Sun was supported in part by the NSF under Award 2007108. This article was presented in part at the 2020 IEEE International Symposium on Information Theory (ISIT) [1] and in part at the 2020 18th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT) [2]. The associate editor coordinating the review of this article and approving it for publication was R. Tandon. *(Corresponding author: Kai Wan.)*

Xiang Zhang and Mingyue Ji are with the Department of Electrical and Computer Engineering, The University of Utah, Salt Lake City, UT 84112 USA (e-mail: xiang.zhang@utah.edu; mingyue.ji@utah.edu).

Kai Wan and Giuseppe Caire are with the Department of Electrical Engineering and Computer Science, Technical University of Berlin, 10587 Berlin, Germany (e-mail: kai.wan@tu-berlin.de; caire@tu-berlin.de).

Hua Sun is with the Department of Electrical Engineering, University of North Texas, Denton, TX 76203 USA (e-mail: hua.sun@unt.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCOMM.2021.3091612.

Digital Object Identifier 10.1109/TCOMM.2021.3091612

[1] The capacity of PIR is defined as the inverse of the minimum load.

to be $\left\lceil \frac{K}{M+1} \right\rceil^{-1}$ [12]. For the case of multiple databases ($N \geq 2$), [12] provided an achievable scheme achieving the load $1 + \frac{1}{N} + \ldots + \frac{1}{N^{\left\lceil \frac{K}{M+1} \right\rceil - 1}}$, which was later shown to be optimal in [18]. For PIR-PSI with a single database, [12] showed that the capacity is $(K - M)^{-1}$, implying that the impact of private side information is equivalent to reducing the message library size by M. This effect is also seen in [19] which showed that the capacity of PIR-PSI with arbitrary number of databases is $\left(1 + \frac{1}{N} + \ldots + \frac{1}{N^{K-M-1}}\right)^{-1}$. Second, the authors in [20]–[23] considered the *user-against-user* privacy model where users are prevented from learning each other's demands. The authors in [20] first formulated the *coded caching with private demands* problem where a shared-link caching network with demand privacy, i.e., any user cannot learn anything about the demands of other users, was considered. The goal is to design efficient delivery schemes such that the communication load is minimized while preserving such privacy. Order optimal schemes were proposed based on the concept of virtual user.

This paper formulates the problem of cache-aided Multiuser PIR (MuPIR), where each of the $K_{\rm u}$ cache-equipped users aims to retrieve a message from N distributed non-colluding databases while preventing any one of them from gaining knowledge about the user demands given that the cached content of the users are known to the databases.[2] The contribution of this paper includes: First, based on the novel idea of *cache-aided interference alignment (CIA)*, we construct cache placement and private delivery schemes achieving the memory-load pairs $\left(\frac{N-1}{2N}, \frac{N+1}{N}\right)$ and $\left(\frac{2(N-1)}{2N-1}, \frac{N+1}{2N-1}\right)$ for the case of $K = 2$ messages, $K_{\rm u} = 2$ users and $N \geq 2$ databases. Different from the existing cache-aided interference alignment schemes in [26]–[28] which were designed for the cache-aided interference channels, the purpose of our proposed private delivery scheme is to let each server send symmetric messages (in order to preserve user demand privacy), each of which contains some uncached and undesired symbols (i.e., interference) for each user. The proposed CIA approach effectively aligns these interference for each user and thus facilitates correct decoding. We prove that the proposed scheme is optimal under the constraint of uncoded cache placement. Computer-aided investigation given in [29] also shows that the proposed schemes are optimal for general cache placement when $N = 2$ and 3. Second, for general system parameters $K, K_{\rm u}$ and N, we propose a *Product Design (PD)* which incorporates the SJ scheme [5] into the MAN coded caching scheme [6]. Interestingly, the load of the proposed design is the product of the loads achieved by these two schemes and is optimal within a factor of 8. Moreover, PD is exactly optimal in the high memory regime. Finally, we characterize the optimal memory-load trade-off for the case of $K = K_{\rm u} = N = 2$ where the users demand distinct messages. It is shown that under the constraint of the distinct demands, the optimal load can be strictly smaller than the case with general demands.

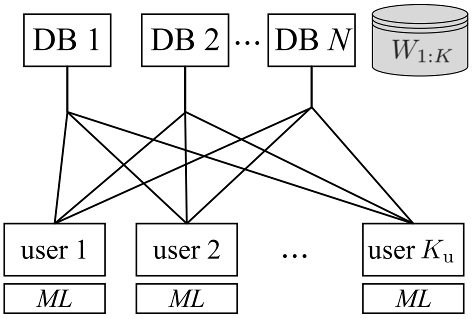


Fig. 1. Cache-aided MuPIR system with N replicated databases, K independent messages and $K_{\rm u}$ cache-equipped users. The users are connected to each DB via an error-free shared-link broadcast channel.

The paper is organized as follows. In Section II, we give a formal description of the problem setup. The main results are given in Section III. In Section IV, we present the proposed CIA based schemes and in Section V, we present the product design for general system parameters. We discuss some interesting observations for the case of distinct demands in Section VI. Finally, we conclude this paper and provide several future directions in Section VII.

Notation Convention: $\mathbb{Z}^+$ denotes the non-negative integer set. $[n] \triangleq \{1, 2, \ldots, n\}$, $[m : n] \triangleq \{m, m+1, \ldots, n\}$ and $(m : n) \triangleq (m, m+1, \ldots, n)$ for some $m \leq n$. For two sets $\mathcal{A}$ and $\mathcal{B}$, the *difference set* is defined as $\mathcal{A} \backslash \mathcal{B} \triangleq \{x \in \mathcal{A} : x \notin \mathcal{B}\}$. For an index set $\mathcal{I}$, denote $A_{\mathcal{I}} \triangleq \{A_i : i \in \mathcal{I}\}$. If $\mathcal{I} = [m : n]$, we write $A_{[m:n]}$ as $A_{m:n}$ for simplicity. For an index vector $I = (i_1, \ldots, i_n)$, denote $A_I \triangleq (A_{i_1}, \ldots, A_{i_n})$. Let $\mathbf{0}_n \triangleq (0, \ldots, 0)$ and $\mathbf{1}_n \triangleq (1, \ldots, 1)$ with length n. $\mathbf{I}_n$ denotes the identity matrix of order n. For a matrix $\mathbf{A}$, $\mathbf{A}(i, :)$ and $\mathbf{A}(:, j)$ denote the i-th row and j-th column of $\mathbf{A}$ respectively. $\mathbf{A}^{\mathsf{T}}$ represents the transpose of $\mathbf{A}$. Operations are on the binary field.

II. Problem Formulation

We consider a system with $K_{\rm u}$ users, each aiming to privately retrieve a message from $N \geq 2$ replicated non-colluding databases (DBs). Each DB stores K independent messages, denoted by $W_1, \ldots, W_K$, each of which is uniformly distributed over $\left[2^L\right]$. Each user is equipped with a cache memory of size $ML (0 \leq M \leq K)$ bits. Let the random variables $Z_1, \ldots, Z_{K_{\rm u}}$ denote the cached content of the users. The system operates in two phases, a *cache placement phase* followed by a *private delivery phase*. In the cache placement phase, the users fill up their cache memory without the knowledge of their future demands. The cached content of each user is a function of $W_{1:K}$ and is assumed to be *known to all DBs*. In the private delivery phase, each user $k \in [K_{\rm u}]$ wishes to retrieve a message W_{θ_k} where $\theta_k \in [K]$. Let $\boldsymbol{\theta} \triangleq (\theta_1, \ldots, \theta_{K_{\rm u}})$ be the demand vector which represents the demands of the users. We assume that $\boldsymbol{\theta}$ follows an arbitrary distribution with full support over $[K]^{K_{\rm u}}$. Depending on $\boldsymbol{\theta}$ and $Z_1, \ldots, Z_{K_{\rm u}}$, the users cooperatively generate N queries

[2] Note that the virtual user strategy and the strategy based on scalar linear function retrieval for coded caching with private demands [20], [24], [25] were designed based on the fact that the user caches are not transparent to each other (but transparent to the single server). However, such an approach which relies on the unawareness of cache to achieve demand privacy cannot be used in the considered MuPIR problem because the databases are aware of the user caches.

$Q_1^{[\boldsymbol{\theta}]}, \ldots, Q_N^{[\boldsymbol{\theta}]}$, and then send the n-th query $Q_n^{[\boldsymbol{\theta}]}$ to DB n. Upon receiving the query, DB n responds with an answer $A_n^{[\boldsymbol{\theta}]}$ broadcasted to all users. The answer $A_n^{[\boldsymbol{\theta}]}$ is a deterministic function of $Q_n^{[\boldsymbol{\theta}]}$, $W_{1:K}$ and $Z_{1:K_{\rm u}}$, which written in terms of conditional entropy, is

$$H\big(A_n^{[\boldsymbol{\theta}]} \big| Q_n^{[\boldsymbol{\theta}]}, W_{1:K}, Z_{1:K_{\rm u}}\big) = 0, \quad \forall n \in [N]. \tag{1}$$

After collecting all the answers from the DBs, the users can correctly recover their desired messages with the help of their cached information, i.e.,

$$H\big(W_{\theta_k} \big| Q_{1:N}^{[\boldsymbol{\theta}]}, A_{1:N}^{[\boldsymbol{\theta}]}, Z_k\big) = 0, \quad \forall k \in [K_{\rm u}]. \tag{2}$$

To preserve the privacy with respect to the DBs, it is required that[3]

$$I\big(\boldsymbol{\theta}; Q_n^{[\boldsymbol{\theta}]}, A_n^{[\boldsymbol{\theta}]}, W_{1:K}, Z_{1:K_{\rm u}}\big) = 0, \quad \forall n \in [N]. \tag{3}$$

Let D denote the total number of bits broadcasted from the DBs, then the *load* of the cache-aided MuPIR problem is defined as[4]

$$R \triangleq \frac{D}{L} = \frac{\sum_{n=1}^{N} H\big(A_n^{[\boldsymbol{\theta}]}\big)}{L}. \tag{4}$$

From the privacy constraint (3), the load can also be written as $R = \frac{1}{L}\sum_{n=1}^{N} H\big(A_n^{[\boldsymbol{\theta}^i]}\big)$, $\forall i \in \big[K^{K_{\rm u}}\big]$ where $\boldsymbol{\theta}^i$ represents the i-th realization of all the $K^{K_{\rm u}}$ possible realizations of the demand vector. This is because the load R should not depend on the user demands $\boldsymbol{\theta}$ otherwise it leaks information about the user demands to the DBs and (3) will be violated.

A memory-load pair (M, R) is said to be achievable if there exists a MuPIR scheme satisfying the decodability constraint (2) and the privacy constraint (3). The goal of the MuPIR problem is to design the cache placement and the corresponding private delivery phases such that the load is minimized. For any $0 \leq M \leq K$, let $R^\star(M)$ denote the minimal achievable load. In addition, if the users directly cache a subset of the library bits, the placement phase is said to be *uncoded*. We denote the minimum achievable load under the constraint of uncoded cache placement by $R^\star_{\rm uncoded}(M)$. Note that any converse bound $R'(M)$ on the worst-case load for the $(K, K_{\rm u})$ coded caching problem without considering demand privacy formulated in [6] is also a converse bound on $R^\star(M)$, i.e., $R^\star(M) = \frac{1}{L}\sum_{n=1}^{N} H(A_n^{[\boldsymbol{\theta}']}) \geq R'(M)$. Similarly, any converse bound $R''(M)$ on the worst-case caching load under the constraint of uncoded cache placement is also a converse on $R^\star_{\rm uncoded}(M)$, i.e., $R^\star_{\rm uncoded}(M) \geq R''(M)$.

III. Main Results

First, we consider the MuPIR problem with $K = K_{\rm u} = 2$ and $N \geq 2$. In this case, we propose a novel *cache-aided interference alignment* (CIA) based scheme (see Section IV) and the corresponding achievable load is given in Theorem 1.

Theorem 1: For the cache-aided MuPIR problem with $K = 2$ messages, $K_{\rm u} = 2$ users and $N \geq 2$ DBs, the following load is achievable

$$R_{\rm CIA}(M) = \begin{cases} 2(1-M), & 0 \leq M \leq \dfrac{N-1}{2N} \\ \dfrac{(N+1)(3-2M)}{2N+1}, & \dfrac{N-1}{2N} \leq M \leq \dfrac{2(N-1)}{2N-1} \\ \left(1-\dfrac{M}{2}\right)\left(1+\dfrac{1}{N}\right), & \dfrac{2(N-1)}{2N-1} \leq M \leq 2 \end{cases} \tag{5}$$

Proof: The proof of Theorem 1 is provided in Section IV, where we present the CIA based approach achieving the memory-load pairs $\big(\frac{N-1}{2N}, \frac{N+1}{N}\big)$ and $\big(\frac{2(N-1)}{2N-1}, \frac{N+1}{2N-1}\big)$. Together with the two trivial pairs $(0, 2)$ and $(2, 0)$, we obtain four corner points. By the memory-sharing among these corner points, the load in Theorem 1 can be achieved. ■

Remark 1: The computer-aided approach given in [29] shows that the achievability result in Theorem 1 is optimal when $N = 2, 3$. For $N \geq 4$, the converse remains open. In addition, the achieved load by the CIA based scheme is better than applying twice the single-user cache-aided PIR of [9] for each user, which yields a load of $(2-M)\big(1+\frac{1}{N}\big) \geq R_{CIA}(M), \forall M \in [0, 2]$.

Corollary 1: The load $R_{\rm CIA}(M)$ in Theorem 1 is optimal when $M \in \big[0, \frac{N-1}{2N}\big] \cup \big[\frac{2(N-1)}{2N-1}, 2\big]$.

Proof: When $M \leq \frac{N-1}{2N}$, $R_{\rm CIA}(M) = 2(1-M)$ coincides with the coded caching converse [6] and therefore is optimal. When $M \geq \frac{2(N-1)}{2N-1}$, $R_{\rm CIA}(M) = \big(1-\frac{M}{2}\big)\big(1+\frac{1}{N}\big)$ coincides with the converse bound $R_{\rm single} = \big(1-\frac{M}{K}\big)\big(1+\frac{1}{N}+\cdots+\frac{1}{N^{K-1}}\big)$ for the single-user cache-aided PIR in [9]. Since increasing the number of users $K_{\rm u}$ cannot decrease the load, the achieved load in Theorem 1 is optimal. ■

Corollary 2: For the cache-aided MuPIR problem with $K = 2$ messages, $K_{\rm u} = 2$ users and $N \geq 2$ DBs, the optimal memory-load trade-off under uncoded cache placement is characterized as

$$R^\star_{\rm uncoded}(M) = \begin{cases} 2 - \dfrac{3}{2}M, & 0 \leq M \leq \dfrac{2(N-1)}{2N-1} \\ \left(1-\dfrac{M}{2}\right)\left(1+\dfrac{1}{N}\right), & \dfrac{2(N-1)}{2N-1} \leq M \leq 2 \end{cases} \tag{6}$$

Proof: For achievability, the corner points in Corollary 2 are the memory-load pairs $(0, 2)$, $\big(\frac{2(N-1)}{2N-1}, \frac{N+1}{2N-1}\big)$, and $(2, 0)$, which can be achieved by the same scheme as Theorem 1. It can be seen in Section IV that the achievable schemes for these corner points are uncoded, and by memory-sharing, the load of Corollary 2 can be achieved.

Under the assumption of uncoded cache placement, as shown in Section II, the converse bound for the shared-link

[3] The privacy constraint (3) can be equivalently written as $I\big(\boldsymbol{\theta}; Q_n^{[\boldsymbol{\theta}]}, W_{1:K}, Z_{1:K}\big) = 0, \forall n \in [N]$ since the answer $A_n^{[\boldsymbol{\theta}]}$ is a deterministic function of $Q_n^{[\boldsymbol{\theta}]}$ and $W_{1:K}$.

[4] Note that in the literature of PIR, R denotes the retrieval *rate*, which is the number of useful bits per bit of download in the single-user case. Our definition of load is consistent with the caching literature, representing the (normalized) number of bits transmitted through the shared-link.

coded caching problem without privacy constraint in [30], [31] is also a converse for our considered cache-aided MuPIR problem. When $0 \le M \le 1$, it was proved that $R^{\star}_{\text{uncoded}}(M) \ge 2 - \frac{3}{2}M$. In addition, by the single-user cache-aided PIR converse in [9], we have $R^{\star}_{\text{uncoded}}(M) \ge \left(1 - \frac{M}{2}\right)\left(1 + \frac{1}{N}\right)$. Since the achievability and converse match, the optimal trade-off in Corollary 2 is characterized. ■

For general K, K_{u} and N, we propose an achievable scheme called *Product Design* (PD). The corresponding achievable load is given by the following theorem.

Theorem 2: The proposed product design achieves the load of $R_{\text{PD}}(M) = \min\left\{K - M, \widehat{R}(M)\right\}$ in which

$$\widehat{R}(M) = \frac{K_{\text{u}} - t}{t+1}\left(1 + \frac{1}{N} + \cdots + \frac{1}{N^{K-1}}\right), \qquad (7)$$

where $t \triangleq \frac{K_{\text{u}}M}{K} \in [1 : K_{\text{u}}]$. When $M = 0$, the load $R_0 \triangleq \min\left\{K_{\text{u}} R_{\text{MPIR}}(K, K_{\text{u}}, N)^{-1}, K\right\}$ is achievable if $K_{\text{u}} < K$, where $R_{\text{MPIR}}(K, K_{\text{u}}, N)$ represents the achievable sum rate in [32] for the Multi-message PIR (MPIR) problem with K messages, N DBs and the user desires K_{u} messages. For non-integer values of t, the lower convex envelope the integer points $(0, R_0)$, $\left(\frac{tK}{K_{\text{u}}}, R_{\text{PD}}(M)\right), t \in [1 : K_{\text{u}}]$ are achievable. Moreover, $\frac{R_{\text{PD}}(M)}{R^{\star}(M)} \le 8$.

Proof:

Achievability: See Section V for the schemes to achieve $\widehat{R}(M)$. The load $K - M$ can be achieved by letting each user store the same $\frac{M}{K}$ fraction of all messages and then downloading the remaining $1 - \frac{M}{K}$ fraction of all messages in the delivery phase, achieving the load $K\left(1 - \frac{M}{K}\right) = K - M$. Each user can recover all the K messages including the desired one. Since the above delivery does not depend on $\boldsymbol{\theta}$, it is private. When $K > K_{\text{u}}$ and $M = 0$, i.e., the users do not have any cache, the considered cache-aided MuPIR problem reduces to the MPIR problem, where a superuser requests K_{u} out K messages without leaking the identities the desired messages. The MPIR scheme proposed in [32] achieving the load R_0, can be used to improve PD based on the observation that *a joint retrieval of multiple messages is better than repeating the single-message retrieval multiple times*. Therefore, the MPIR scheme of [32] can be used when $M = 0$ to achieve the load $R_0 = K_{\text{u}} R_{\text{MPIR}}(K, K_{\text{u}}, N)^{-1}$, which is strictly better than repeating the single-message retrieval scheme K_{u} times by plugging $t = 0$ into (7).

Converse: We use the converse bound in [33] on the worst-case load for coded caching without privacy constraint, denoted by $R_{\text{caching}}(M)$. As shown in Section II, $R_{\text{caching}}(M)$ is also a converse bound for the considered MuPIR problem. In addition, it was proved in [33] that $R_{\text{caching}}(M)$ is no less than the lower convex envelope of $\frac{1}{4}\min\left\{\frac{K_{\text{u}}-t}{t+1}, K\left(1 - \frac{M}{K}\right)\right\}$ where $t \in [K_{\text{u}}]$. Hence, we have

$$\frac{R_{\text{PD}}(M)}{R^{\star}(M)} \le \frac{R_{\text{PD}}(M)}{R_{\text{caching}}(M)} \le 4 \cdot \left(1 + \frac{1}{N} + \cdots + \frac{1}{N^{K-1}}\right) \le 8, \qquad (8)$$

which is due to $R_{\text{PD}}(M) = \min\left\{K\left(1 - \frac{M}{K}\right), \widehat{R}(M)\right\}$ and $1 + \frac{1}{N} + \cdots + \frac{1}{N^{K-1}} \le 2, \forall N \ge 2$. ■

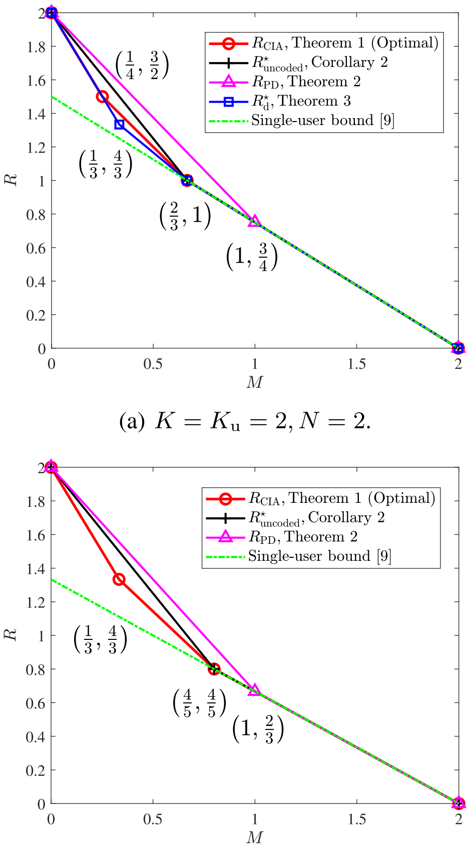
Corollary 3: The proposed product design is optimal when $M \in \left[\frac{(K_{\text{u}}-1)}{K_{\text{u}}}K, K\right]$.

Proof: When $K \ge K_{\text{u}}$ and $M = \frac{K(K_{\text{u}}-1)}{K_{\text{u}}}$, (7) becomes $\widehat{R}(M) = \frac{1}{K_{\text{u}}}\left(1 + \cdots + \frac{1}{N^{K-1}}\right)$. On the other hand, the author in [9] showed that when $K_{\text{u}} = 1$, the optimal single-user cache-aided PIR load is equal to $\left(1 - \frac{M}{K}\right)\left(1 + \cdots + \frac{1}{N^{K-1}}\right)$, which also equals (7) when $M = \frac{K(K_{\text{u}}-1)}{K_{\text{u}}}$. By the memory-sharing between $\left(\frac{K(K_{\text{u}}-1)}{K_{\text{u}}}, \widehat{R}\left(\frac{K(K_{\text{u}}-1)}{K_{\text{u}}}\right)\right)$ and $(K, 0)$, we conclude that PD is optimal when $\frac{(K_{\text{u}}-1)}{K_{\text{u}}}K \le M \le K$. ■

Numerical Evaluation: In Fig. 2, we consider the MuPIR systems with $K = K_u = 2$, where $N = 2$ in Fig. 2a and $N = 3$ in Fig. 2b respectively. We compare the proposed CIA based scheme in Theorem 1, the optimal scheme under uncoded cache placement in Corollary 6, the product design in Theorem 2, and the computer-aided converse in [29]. In addition, it will be clarified in Theorem 3 that for the case $K = K_u = N = 2$, if the users demand distinct messages, the optimal memory-load trade-off is $R^{\star}_{\text{d}}(M)$ given in (43). In Fig. 2a, there are two non-trivial corner points $(1/4, 3/2)$ and $(2/3, 1)$ associated with the CIA based scheme in Theorem 1. It can be seen that in the case of general demands, the CIA based scheme outperforms both the optimal scheme under uncoded cache placement in Corollary 2 and the product design in Theorem 2. When $1/4 \le M \le 2$, $R_{\text{CIA}}(M)$ coincides with the computer-aided converse [29] and hence is optimal. It also can be seen that a lower load can be achieved when users only have distinct demands. Fig. 2b shows the case when $N = 3$ in which $R_{\text{CIA}}(M)$ is optimal when $1/3 \le M \le 2$ by the computer-aided converse. In Fig. 3, we compare the load of the product design with the best-known caching bound provided in [8] when $K = K_{\text{u}} = 6, N = 2$. More specifically, Theorem 2 of [8] gives the caching converse as a lower convex envelope of the set of memory-load pairs $\left\{\left(\frac{7-\ell}{s}, \frac{s-1}{2} + \frac{\ell(\ell-1)}{2s}\right) : \forall s \in [1 : 6], \forall \ell \in [1 : s]\right\} \cup \{(0, 6)\}$. Since $R_{\text{PD}}(1)$ lies above the line segment connecting the memory-load pairs $(0, 6)$ and $(2, R_{\text{PD}}(2))$, we can use memory-sharing to achieve a better load for $M = 1$.

IV. Proof of Theorem 1: Description of the CIA Scheme

In this section, we prove Theorem 1. For the cache-aided MuPIR problem with $K = 2$ messages, $K_{\text{u}} = 2$ users and $N \ge 2$ DBs, we first show the achievability of the memory-load pairs $\left(\frac{N-1}{2N}, \frac{N+1}{N}\right)$ and $\left(\frac{2(N-1)}{2N-1}, \frac{N+1}{2N-1}\right)$ using the proposed CIA scheme. Note that when $M = 0$, we let any one of the DBs broadcast the two messages to the users, so the memory-load pair $(0, 2)$ is achievable. When $M = 2$, we let both users store the two messages in the placement phase and there is no need for the DBs to transmit anything, implying that $(2, 0)$ is achievable. By the memory-sharing between the above four corner points, the load of Theorem 1 can be achieved. For each of the above two non-trivial corner points, we first describe the general achievable schemes for arbitrary number of DBs and then present an example to highlight the design



(a) $K = K_{\rm u} = 2, N = 2$.

(b) $K = K_{\rm u} = 2, N = 3$

Fig. 2. An illustration of the achievable load R of the MuPIR system with $K = K_u = 2$. (a) $N = 2$. Both the CIA scheme ($R_{\rm CIA}$) and the optimal scheme under distinct demands ($R^\star_{\rm d}(M)$) have four corner pints; Both the optimal scheme under uncoded cache placement ($R^\star_{\rm uncoded}$) and the product design ($R_{\rm PD}$) have three corner points; (b) $N = 3$. $R_{\rm CIA}$ has four corner points. $R^\star_{\rm uncoded}$ and $R_{\rm PD}$ have three corner points.

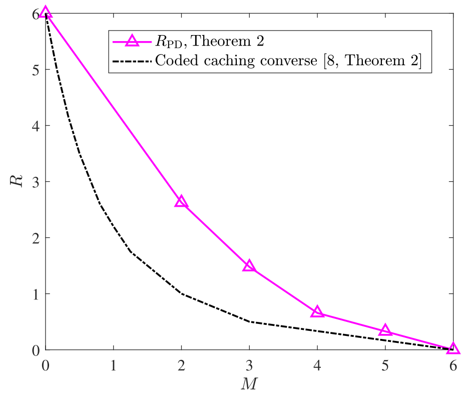


Fig. 3. The load of the product design compared to the best-known caching bound [8] for $K = K_{\rm u} = 6, N = 2$.

intuition. Computer-aided investigation [29] shows that the achievable load in Theorem 1 is optimal when $N = 2$ and 3. For general values of N, the converse remains open.

A. Achievability of $\left(\frac{N-1}{2N},\frac{N+1}{N}\right)$

Let $W_1 = A$ and $W_2 = B$ denote the two messages, each consisting of $L = 2N$ bits, i.e., $A = (A_1, \dots, A_{2N}), B = (B_1, \dots, B_{2N})$. The proposed scheme is described as follows.

1) Cache Placement: Each user stores $N-1$ linear combinations of the message bits in its cache (therefore $M = \frac{N-1}{2N}$), i.e.,

$$Z_1 = \left\{\boldsymbol{\alpha}_{1,j} A^{\mathsf T}_{(1:N)} + \boldsymbol{\beta}_{1,j} B^{\mathsf T}_{(1:N)} : j \in [N-1]\right\}, \tag{9a}$$

$$Z_2 = \left\{\boldsymbol{\alpha}_{2,j} A^{\mathsf T}_{(N+1:2N)} + \boldsymbol{\beta}_{2,j} B^{\mathsf T}_{(N+1:2N)} : j \in [N-1]\right\}, \tag{9b}$$

in which the linear combination coefficients $\boldsymbol{\alpha}_{i,j}, \boldsymbol{\beta}_{i,j} \in \mathbb{F}_2^{1\times N}\backslash\{\mathbf{0}_N\}, \forall i \in [2], \forall j \in [N-1]$ are chosen such that $\text{rank}([\boldsymbol{\alpha}_{i,1}; \dots; \boldsymbol{\alpha}_{i,N-1}]) = N-1$ and $\text{rank}([\boldsymbol{\beta}_{i,1}; \dots; \boldsymbol{\beta}_{i,N-1}]) = N-1$, $\forall i \in [2]$. WLOG, we choose the coefficients to be $[\boldsymbol{\alpha}_{i,1}; \dots; \boldsymbol{\alpha}_{i,N-1}] = [\boldsymbol{\beta}_{i,1}; \dots; \boldsymbol{\beta}_{i,N-1}] = \left[\mathbf{I}_{N-1}, \mathbf{0}^{\mathsf T}_{N-1}\right], \forall i \in [2]$. Recall that $A_{(1:N)} \triangleq (A_1, \dots, A_N)$ and other notations follow similarly. Furthermore, let $Z_{i,j}$ denote the j-th linear combination in Z_i, i.e., $Z_{1,j} = \boldsymbol{\alpha}_{1,j} A^{\mathsf T}_{(1:N)} + \boldsymbol{\beta}_{1,j} B^{\mathsf T}_{(1:N)}$, $Z_{2,j} = \boldsymbol{\alpha}_{2,j} A^{\mathsf T}_{(N+1:2N)} + \boldsymbol{\beta}_{2,j} B^{\mathsf T}_{(N+1:2N)}, \forall j \in [N-1], \forall i \in [2]$.

2) Private Delivery: In this phase, the two users download an answer from each DB according to their demands (θ_1, θ_2). The answers are in the form of random linear combinations of certain message bits. In particular, the answer of DB $n \in [N-1]$ consists of two random linear combinations, i.e., $A_n^{[\boldsymbol{\theta}]} \triangleq \left(A_{n,1}^{[\boldsymbol{\theta}]}, A_{n,2}^{[\boldsymbol{\theta}]}\right)$ where $A_{n,1}^{[\boldsymbol{\theta}]} = \mathbf{u}_{n,1} A^{\mathsf T}_{(1:N)} + \mathbf{v}_{n,1} B^{\mathsf T}_{(1:N)}, A_{n,2}^{[\boldsymbol{\theta}]} = \mathbf{u}_{n,2} A^{\mathsf T}_{(N+1:2N)} + \mathbf{v}_{n,2} B^{\mathsf T}_{(N+1:2N)}$. The answer of DB N consists of four random linear combinations, i.e., $A_n^{[\boldsymbol{\theta}]} \triangleq \left(A_{N,1}^{[\boldsymbol{\theta}]}, A_{N,2}^{[\boldsymbol{\theta}]}, A_{N,3}^{[\boldsymbol{\theta}]}, A_{N,4}^{[\boldsymbol{\theta}]}\right)$ where $A_{N,1}^{[\boldsymbol{\theta}]} = \mathbf{g}_1 A^{\mathsf T}_{(1:N)}$, $A_{N,2}^{[\boldsymbol{\theta}]} = \mathbf{g}_2 B^{\mathsf T}_{(1:N)}$, $A_{N,3}^{[\boldsymbol{\theta}]} = \mathbf{g}_3 A^{\mathsf T}_{(N+1:2N)}$ and $A_{N,4}^{[\boldsymbol{\theta}]} = \mathbf{g}_4 B^{\mathsf T}_{(N+1:2N)}$. The coefficient vectors $\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3, \mathbf{g}_4, \mathbf{u}_{n,j}, \mathbf{v}_{n,j} \in \mathbb{F}_2^{1\times N}, \forall n \in [N-1], \forall j \in [2]$ used in the answers are subject to design according to the user demands. Therefore, $2N+2$ linear combinations will be downloaded in total in the delivery phase. The answers can be written as

$$\begin{bmatrix} A_{1,1}^{[\boldsymbol{\theta}]} \\ A_{1,2}^{[\boldsymbol{\theta}]} \\ \vdots \\ A_{N-1,1}^{[\boldsymbol{\theta}]} \\ A_{N-1,2}^{[\boldsymbol{\theta}]} \\ A_{N,1}^{[\boldsymbol{\theta}]} \\ A_{N,2}^{[\boldsymbol{\theta}]} \\ A_{N,3}^{[\boldsymbol{\theta}]} \\ A_{N,4}^{[\boldsymbol{\theta}]} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_{1,1} & \mathbf{0}_N & \mathbf{v}_{1,1} & \mathbf{0}_N \\ \mathbf{0}_N & \mathbf{u}_{1,2} & \mathbf{0}_N & \mathbf{v}_{1,2} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{u}_{N-1,1} & \mathbf{0}_N & \mathbf{v}_{N-1,1} & \mathbf{0}_N \\ \mathbf{0}_N & \mathbf{u}_{N-1,2} & \mathbf{0}_N & \mathbf{v}_{N-1,2} \\ \mathbf{g}_1 & \mathbf{0}_N & \mathbf{0}_N & \mathbf{0}_N \\ \mathbf{0}_N & \mathbf{0}_N & \mathbf{g}_2 & \mathbf{0}_N \\ \mathbf{0}_N & \mathbf{g}_3 & \mathbf{0}_N & \mathbf{0}_N \\ \mathbf{0}_N & \mathbf{0}_N & \mathbf{0}_N & \mathbf{g}_4 \end{bmatrix} \times \begin{bmatrix} A^{\mathsf T}_{(1:N)} \\ A^{\mathsf T}_{(N+1:2N)} \\ B^{\mathsf T}_{(1:N)} \\ B^{\mathsf T}_{(N+1:2N)} \end{bmatrix}. \tag{10}$$

We next show how the linear coefficients can be designed using the idea of CIA such that the users can correctly recover their desired messages. Due to space limit, we will only consider $(\theta_1, \theta_2) = (1, 2)$ and $(1, 1)$. Other cases work similarly and are omitted here.

For $(\theta_1, \theta_2) = (1, 2)$, i.e., user 1 and 2 demand messages A and B respectively, the following six N-by-N coefficient matrices should be full-rank:

Full-rank condition: the following matrices are full-rank,

$$\underbrace{\begin{bmatrix} \boldsymbol{\alpha}_{1,1} \\ \vdots \\ \boldsymbol{\alpha}_{1,N-1} \\ \mathbf{g}_1 \end{bmatrix}, \begin{bmatrix} \boldsymbol{\beta}_{2,1} \\ \vdots \\ \boldsymbol{\beta}_{2,N-1} \\ \mathbf{g}_4 \end{bmatrix}, \begin{bmatrix} \mathbf{u}_{1,2} \\ \vdots \\ \mathbf{u}_{N-1,2} \\ \mathbf{g}_3 \end{bmatrix}, \begin{bmatrix} \mathbf{v}_{1,1} \\ \vdots \\ \mathbf{v}_{N-1,1} \\ \mathbf{g}_2 \end{bmatrix},}_{\text{For correct decoding of } (\theta_1,\theta_2)=(1,2)}$$
$$\underbrace{\begin{bmatrix} \boldsymbol{\alpha}_{2,1} \\ \vdots \\ \boldsymbol{\alpha}_{2,N-1} \\ \mathbf{g}_3 \end{bmatrix}, \begin{bmatrix} \boldsymbol{\beta}_{1,1} \\ \vdots \\ \boldsymbol{\beta}_{1,N-1} \\ \mathbf{g}_2 \end{bmatrix}}_{\text{For privacy}}. \tag{11}$$

Note that only the first four coefficient matrices being full-rank in (11) is mandatory for the decoding of the demands $(\theta_1, \theta_2) = (1, 2)$. The two extra matrices $[\boldsymbol{\alpha}_{2,1}; \ldots; \boldsymbol{\alpha}_{2,N-1}; \mathbf{g}_3]$ and $[\boldsymbol{\beta}_{1,1}; \ldots; \boldsymbol{\beta}_{1,N-1}; \mathbf{g}_2]$ being full-rank is mandatory for the decoding of other user demands. The reason that we require the two extra matrices to be full-rank for $(\theta_1, \theta_2) = (1, 2)$ is that, if the two matrices are not full-rank here (the DBs can check this since the users' cache coefficients $\boldsymbol{\alpha}_{2,n}, \boldsymbol{\beta}_{1,n}, \forall n \in [N-1]$ are known to the DBs), then DB 2 can know that the actual demands being requested are $(\theta_1, \theta_2) = (1, 2)$ since correct decoding is impossible for any demands other than $(1, 2)$. In fact, to preserve privacy, any full-rank coefficient matrix consisting of the linear coefficients of one DB and the cache coefficients which are necessary for the correct decoding of one demand vector (θ_1, θ_2) must be full-rank for all possible demands $(\theta_1, \theta_2) \in [2]^2$. This multi-purpose full-rank requirement holds for all user demands. The required alignment is

$$\textit{Alignment condition}: \mathbf{g}_1 = \mathbf{u}_{n,1}, \quad \forall n \in [N-1], \tag{12a}$$
$$\mathbf{g}_4 = \mathbf{v}_{n,2}, \quad \forall n \in [N-1]. \tag{12b}$$

We next show that with the above full-rank and alignment conditions, the two users can correctly recover the messages A and B respectively.

Due to the alignment condition of (12b), we have $A_{N,4}^{[(1,2)]} = \mathbf{g}_4 B_{(N+1:2N)}^{\mathsf{T}} = \mathbf{v}_{n,2} B_{(N+1:2N)}^{\mathsf{T}}, \forall n \in [N-1]$, i.e., the message bits $B_{(N+1:2N)}$ are aligned among the linear combinations $A_{1,2}^{[(1,2)]}, \ldots, A_{N-1,2}^{[(1,2)]}$. Subtracting $A_{N,4}^{[(1,2)]}$ from $A_{1,2}^{[(1,2)]}, \ldots, A_{N-1,2}^{[(1,2)]}$ in (10), we obtain

$$\begin{bmatrix} A_{1,2}^{[(1,2)]} - A_{N,4}^{[(1,2)]} \\ \vdots \\ A_{N-1,2}^{[(1,2)]} - A_{N,4}^{[(1,2)]} \\ A_{N,3}^{[(1,2)]} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_{1,2} \\ \vdots \\ \mathbf{u}_{N-1,2} \\ \mathbf{g}_3 \end{bmatrix} A_{(N+1:2N)}^{\mathsf{T}}. \tag{13}$$

By (11), the coefficient matrix on the RHS of (13) is full-rank. Therefore, both users can decode $A_{(N+1:2N)}$ as

$$A_{(N+1:2N)}^{\mathsf{T}} = \begin{bmatrix} \mathbf{u}_{1,2} \\ \vdots \\ \mathbf{u}_{N-1,2} \\ \mathbf{g}_3 \end{bmatrix}^{-1} \begin{bmatrix} A_{1,2}^{[(1,2)]} - A_{N,4}^{[(1,2)]} \\ \vdots \\ A_{N-1,2}^{[(1,2)]} - A_{N,4}^{[(1,2)]} \\ A_{N,3}^{[(1,2)]} \end{bmatrix}. \tag{14}$$

Similarly, due to the alignment condition of (12a), we have $A_{N,1}^{[(1,2)]} = \mathbf{g}_1 A_{(1:N)}^{\mathsf{T}} = \mathbf{u}_{n,1} A_{(1:N)}^{\mathsf{T}}, \forall n \in [N-1]$, i.e., $A_{(1:N)}$ are aligned among $A_{1,1}^{[(1,2)]}, \ldots, A_{N-1,1}^{[(1,2)]}$. Subtracting $A_{N,1}^{[(1,2)]}$ from $A_{1,1}^{[(1,2)]}, \ldots, A_{N-1,1}^{[(1,2)]}$, and by the full-rank condition (11), $B_{(1:N)}$ can be decoded by both users as

$$B_{(1:N)}^{\mathsf{T}} = \begin{bmatrix} \mathbf{v}_{1,1} \\ \vdots \\ \mathbf{v}_{N-1,1} \\ \mathbf{g}_2 \end{bmatrix}^{-1} \begin{bmatrix} A_{1,1}^{[(1,2)]} - A_{N,1}^{[(1,2)]} \\ \vdots \\ A_{N-1,1}^{[(1,2)]} - A_{N,1}^{[(1,2)]} \\ A_{N,2}^{[(1,2)]} \end{bmatrix}. \tag{15}$$

Now the message bits $A_{(N+1:2N)}, B_{(1:N)}$ are available to both users. User 1 still needs $A_{(1:N)}$ and user 2 still needs $B_{(N+1:2N)}$. Removing the interference of $B_{(1:N)}$ from Z_1, user 1 obtains $N-1$ linear combinations of $A_{(1:N)}$. Together with $A_{N,1}^{[(1,2)]} = \mathbf{g}_1 A_{(1:N)}^{\mathsf{T}}$, user 1 obtains N independent linear combinations of $A_{(1:N)}$, from which $A_{(1:N)}$ can be decoded as

$$A_{(1:N)}^{\mathsf{T}} = \begin{bmatrix} \boldsymbol{\alpha}_{1,1} \\ \vdots \\ \boldsymbol{\alpha}_{1,N-1} \\ \mathbf{g}_1 \end{bmatrix}^{-1} \begin{bmatrix} Z_{1,1} - \boldsymbol{\beta}_{1,1} B_{(1:N)}^{\mathsf{T}} \\ \vdots \\ Z_{1,N-1} - \boldsymbol{\beta}_{1,N-1} B_{(1:N)}^{\mathsf{T}} \\ A_{N,1}^{[(1,2)]} \end{bmatrix}. \tag{16}$$

As a result, user 1 correctly decodes all the $2N$ bits of the desired message A. Similarly, user 2 can also correctly decode all the $2N$ bits of message B.

For $(\theta_1, \theta_2) = (1, 1)$, the following six coefficient matrices

$$\begin{bmatrix} \boldsymbol{\alpha}_{1,1} \\ \boldsymbol{\alpha}_{1,2} \\ \vdots \\ \boldsymbol{\alpha}_{1,N-1} \\ \mathbf{g}_1 \end{bmatrix}, \begin{bmatrix} \boldsymbol{\alpha}_{2,1} \\ \boldsymbol{\alpha}_{2,2} \\ \vdots \\ \boldsymbol{\alpha}_{2,N-1} \\ \mathbf{g}_3 \end{bmatrix}, \begin{bmatrix} \boldsymbol{\beta}_{1,1} \\ \boldsymbol{\beta}_{1,2} \\ \vdots \\ \boldsymbol{\beta}_{1,N-1} \\ \mathbf{g}_2 \end{bmatrix}, \begin{bmatrix} \boldsymbol{\beta}_{2,1} \\ \boldsymbol{\beta}_{2,2} \\ \vdots \\ \boldsymbol{\beta}_{2,N-1} \\ \mathbf{g}_4 \end{bmatrix},$$
$$\times \begin{bmatrix} \mathbf{u}_{1,1} \\ \mathbf{u}_{2,1} \\ \vdots \\ \mathbf{u}_{N-1,1} \\ \mathbf{g}_1 \end{bmatrix}, \begin{bmatrix} \mathbf{u}_{1,2} \\ \mathbf{u}_{2,2} \\ \vdots \\ \mathbf{u}_{N-1,2} \\ \mathbf{g}_3 \end{bmatrix} \tag{17}$$

are required to be full-rank. The alignment condition is

$$\mathbf{g}_2 = \mathbf{v}_{n,1}, \quad \forall n \in [N-1], \tag{18a}$$
$$\mathbf{g}_4 = \mathbf{v}_{n,2}, \quad \forall n \in [N-1]. \tag{18b}$$

With the above conditions, we next show that both users can correctly recover message A.

Due to (18a), we have $A_{N,2}^{[(1,1)]} = \mathbf{g}_2 B_{(1:N)}^{\mathsf{T}} = \mathbf{v}_{n,1} B_{(1:N)}^{\mathsf{T}}, \forall n \in [N-1]$. Subtracting $A_{N,2}^{[(1,1)]}$ from $A_{n,1}^{[(1,1)]}, \ldots, A_{N-1,1}^{[(1,1)]}$, and by the full-rank condition (17), both users can decode $A_{(1:N)}$ as

$$A_{(1:N)}^{\mathsf{T}} = \begin{bmatrix} \mathbf{u}_{1,1} \\ \mathbf{u}_{2,1} \\ \vdots \\ \mathbf{u}_{N-1,1} \\ \mathbf{g}_1 \end{bmatrix}^{-1} \begin{bmatrix} A_{1,1}^{[(1,1)]} - A_{N,2}^{[(1,1)]} \\ A_{2,1}^{[(1,1)]} - A_{N,2}^{[(1,1)]} \\ \vdots \\ A_{N-1,1}^{[(1,1)]} - A_{N,2}^{[(1,1)]} \\ A_{N,1}^{[(1,1)]} \end{bmatrix}. \tag{19}$$

Also, due to (18b), we have $A_{N,4}^{[(1,1)]} = \mathbf{g}_4 B_{(N+1:2N)}^{\mathsf{T}} = \mathbf{v}_{n,2} B_{(N+1:2N)}^{\mathsf{T}}, \forall n \in [N-1]$. Subtracting $A_{N,4}^{[(1,1)]}$ from $A_{n,2}^{[(1,1)]}, \ldots, A_{N-1,2}^{[(1,1)]}$, and by (17), both users can decode $A_{(N+1:2N)}$ as

$$A_{(N+1:2N)}^{\mathsf{T}} = \begin{bmatrix} \mathbf{u}_{1,2} \\ \vdots \\ \mathbf{u}_{N-1,2} \\ \mathbf{g}_3 \end{bmatrix}^{-1} \begin{bmatrix} A_{1,2}^{[(1,1)]} - A_{N,4}^{[(1,1)]} \\ \vdots \\ A_{N-1,2}^{[(1,1)]} - A_{N,4}^{[(1,1)]} \\ A_{N,3}^{[(1,1)]} \end{bmatrix}. \tag{20}$$

As a result, both users correctly recover message A.

Remark 2: One interesting observation is that, for the case of identical demands, i.e., $(\theta_1, \theta_2) = (1,1)$ *or* $(2,2)$, *the cached contents of the users are actually not used in the decoding process. This means that caching does not help to improve the PIR load in the case of identical demands for the considered problem setup* $K = K_{\mathrm{u}} = 2$.

With the above full-rank and alignment conditions, we now employ a randomized specification of the linear combination coefficients used by each DB and formally describe the delivery scheme.

We first introduce some necessary notations. Define a binary matrix $\mathbf{Y}_N \in \mathbb{F}_2^{N \times N} (N \geq 2)$ as

$$\mathbf{Y}_N \triangleq \begin{bmatrix} \mathbf{I}_{N-1} & \mathbf{1}_{N-1}^{\mathsf{T}} \\ \mathbf{0}_{N-1} & 1 \end{bmatrix} \tag{21}$$

It can be seen that $\mathrm{rank}(\mathbf{Y}_N) = N$. Let $\mathcal{Y}_N \triangleq \{\mathbf{Y}_N(i,:) : i \in [N]\}$ be a set that contains the rows of $\mathbf{Y}_N$. Also define two binary matrices $\mathbf{M}(\mathbf{u}_{:,i}, \mathbf{g}_j), \mathbf{M}(\mathbf{v}_{:,i}, \mathbf{g}_j) \in \mathbb{F}_2^{N \times N}$, $\forall i \in [2], \forall j \in [4]$ as

$$\mathbf{M}(\mathbf{u}_{:,i}, \mathbf{g}_j) \triangleq [\mathbf{u}_{1,i}; \mathbf{u}_{2,i}; \ldots; \mathbf{u}_{N-1,i}; \mathbf{g}_j], \tag{22a}$$

$$\mathbf{M}(\mathbf{v}_{:,i}, \mathbf{g}_j) \triangleq [\mathbf{v}_{1,i}; \mathbf{v}_{2,i}; \ldots; \mathbf{v}_{N-1,i}; \mathbf{g}_j]. \tag{22b}$$

Note that $\mathbf{M}(\mathbf{u}_{:,i}, \mathbf{g}_j)$ and $\mathbf{M}(\mathbf{v}_{:,i}, \mathbf{g}_j)$ represent the coefficient sub-matrices of (10) corresponding to the message bits $A_{(i-1)N+1:iN}$ and $B_{(i-1)N+1:iN}$ respectively.

The delivery strategies for different user demands are given as follows:

- $(\theta_1, \theta_2) = (1,2)$: Let $\mathbf{g}_1$ and $\mathbf{g}_4$ be chosen randomly and uniformly i.i.d. from $\mathcal{Y}_N$. Also let $\mathbf{M}(\mathbf{u}_{:,2}, \mathbf{g}_3)$ and $\mathbf{M}(\mathbf{v}_{:,1}, \mathbf{g}_2)$ be two independent random permutations of the rows of $\mathbf{Y}_N$. It can be seen that with such a specification of the answer coefficients and the previously defined cache coefficients, the full-rank and the alignment conditions (11), (12) are satisfied. Therefore, both users can recover their desired messages.
- $(\theta_1, \theta_2) = (1,1)$: Let $\mathbf{g}_2$ and $\mathbf{g}_4$ be chosen randomly and uniformly i.i.d. from $\mathcal{Y}_N$. Also let $\mathbf{M}(\mathbf{u}_{:,1}, \mathbf{g}_1)$ and $\mathbf{M}(\mathbf{u}_{:,2}, \mathbf{g}_3)$ be two independent random permutations of the rows of $\mathbf{Y}_N$. It can be seen that the full-rank and alignment conditions (17), (18) are satisfied.
- $(\theta_1, \theta_2) = (2,1)$: Let $\mathbf{g}_2$ and $\mathbf{g}_3$ be chosen randomly and uniformly i.i.d. from $\mathcal{Y}_N$. Also let $\mathbf{M}(\mathbf{u}_{:,1}, \mathbf{g}_1)$ and $\mathbf{M}(\mathbf{u}_{:,2}, \mathbf{g}_4)$ be two independent random permutations of the rows of $\mathbf{Y}_N$. It can be verified that the corresponding full-rank and alignment conditions are satisfied.
- $(\theta_1, \theta_2) = (2,2)$: Let $\mathbf{g}_1$ and $\mathbf{g}_3$ be chosen randomly and uniformly i.i.d. from $\mathcal{Y}_N$. Also let $\mathbf{M}(\mathbf{v}_{:,1}, \mathbf{g}_2)$ and $\mathbf{M}(\mathbf{v}_{:,2}, \mathbf{g}_4)$ be two independent random permutations of the rows of $\mathbf{Y}_N$.

We next prove the correctness and privacy of the above delivery scheme.

Correctness: Since the random specification of the answer coefficients satisfies the corresponding full-rank and alignment conditions for any $(\theta_1, \theta_2) \in [2]^2$, decodability is guaranteed.

Privacy: WLOG, we prove that the above delivery scheme is private from DB 1's viewpoint, i.e., the demand vector $\boldsymbol{\theta}$ is equally likely to be $(1,2), (2,1), (1,1)$ or $(2,2)$. More specifically, let $\boldsymbol{x} \triangleq [\mathbf{u}_{1,1}, \mathbf{u}_{1,2}, \mathbf{v}_{1,1}, \mathbf{v}_{1,2}] \in \mathcal{Y}_N^{1 \times 4}$ be a random realization of the answer linear coefficients of DB 1. Let $\boldsymbol{\Gamma}(\mathbf{u}_{1,j}, \boldsymbol{\theta})$ denote a random query of the value of $\mathbf{u}_{1,j}, j = 1,2$ to DB 1 when the demand vector is $\boldsymbol{\theta}$. Other notations follow similarly. Let $\mathbf{X}(\boldsymbol{\theta}) \triangleq [\boldsymbol{\Gamma}(\mathbf{u}_{1,1}, \boldsymbol{\theta}), \boldsymbol{\Gamma}(\mathbf{u}_{1,2}, \boldsymbol{\theta}), \boldsymbol{\Gamma}(\mathbf{v}_{1,1}, \boldsymbol{\theta}), \boldsymbol{\Gamma}(\mathbf{v}_{1,2}, \boldsymbol{\theta})]$ represent the random query to DB 1 when the user demand vector is $\boldsymbol{\theta}$. Then the probability that $\boldsymbol{x}$ is generated for $\boldsymbol{\theta}$, i.e., $\mathbf{X}(\boldsymbol{\theta}) = \boldsymbol{x}$, is equal to

$$\begin{aligned} &P\left(\mathbf{X}(\boldsymbol{\theta}) = \boldsymbol{x}\right) \\ &\quad = P\left(\boldsymbol{\Gamma}(\mathbf{u}_{1,1}, \boldsymbol{\theta}) = \mathbf{u}_{1,1}\right) \times P\left(\boldsymbol{\Gamma}(\mathbf{u}_{1,2}, \boldsymbol{\theta}) = \mathbf{u}_{1,2}\right) \\ &\qquad \times P\left(\boldsymbol{\Gamma}(\mathbf{v}_{1,1}, \boldsymbol{\theta}) = \mathbf{v}_{1,1}\right) \times P\left(\boldsymbol{\Gamma}(\mathbf{v}_{1,2}, \boldsymbol{\theta}) = \mathbf{v}_{1,2}\right) \end{aligned} \tag{23a}$$

$$= \begin{cases} \dfrac{1}{N} \times \dfrac{(N-1)!}{N!} \times \dfrac{(N-1)!}{N!} \times \dfrac{1}{N} = \dfrac{1}{N^4}, \\ \quad \text{if } \boldsymbol{\theta} = (1,2) \\ \dfrac{(N-1)!}{N!} \times \dfrac{1}{N} \times \dfrac{1}{N} \times \dfrac{(N-1)!}{N!} = \dfrac{1}{N^4}, \\ \quad \text{if } \boldsymbol{\theta} = (2,1) \\ \dfrac{(N-1)!}{N!} \times \dfrac{(N-1)!}{N!} \times \dfrac{1}{N} \times \dfrac{1}{N} = \dfrac{1}{N^4}, \\ \quad \text{if } \boldsymbol{\theta} = (1,1) \\ \dfrac{1}{N} \times \dfrac{1}{N} \times \dfrac{(N-1)!}{N!} \times \dfrac{(N-1)!}{N!} = \dfrac{1}{N^4}, \\ \quad \text{if } \boldsymbol{\theta} = (2,2) \end{cases} \tag{23b}$$

in which (23a) is because the specifications of the query vectors $\boldsymbol{\Gamma}(\mathbf{u}_{1,1},\boldsymbol{\theta}), \boldsymbol{\Gamma}(\mathbf{u}_{1,2},\boldsymbol{\theta}), \boldsymbol{\Gamma}(\mathbf{v}_{1,1},\boldsymbol{\theta})$ and $\boldsymbol{\Gamma}(\mathbf{v}_{1,2},\boldsymbol{\theta})$ are independent of each other according to the delivery design. Moreover, for $\boldsymbol{\theta}=(1,2)$, because $\mathbf{u}_{1,1}$ and $\mathbf{v}_{1,2}$ are chosen i.i.d. from $\mathcal{Y}_N$, we have $P(\boldsymbol{\Gamma}(\mathbf{u}_{1,1},\boldsymbol{\theta})=\mathbf{u}_{1,1}) = P(\boldsymbol{\Gamma}(\mathbf{v}_{1,2},\boldsymbol{\theta})=\mathbf{v}_{1,2})=\frac{1}{N}$. Also, because $\mathbf{M}(\mathbf{u}_{:,2},\mathbf{g}_3)$ and $\mathbf{M}(\mathbf{v}_{:,1},\mathbf{g}_2)$ are two independent random permutations of $\mathbf{Y}_N$, we have $P(\boldsymbol{\Gamma}(\mathbf{u}_{1,2},\boldsymbol{\theta})=\mathbf{u}_{1,2}) = P(\boldsymbol{\Gamma}(\mathbf{v}_{1,1},\boldsymbol{\theta})=\mathbf{v}_{1,1}) = \frac{(N-1)!}{N!}=\frac{1}{N}$. Therefore, we have $P\left(\mathbf{X}(\boldsymbol{\theta})=\boldsymbol{x}\right)=\frac{1}{N^4}$ if $\boldsymbol{\theta}=(1,2)$. The probabilities for other demands can be calculated similarly. Since $P(\mathbf{X}(\boldsymbol{\theta})=\boldsymbol{x})$ does not depend on $\boldsymbol{\theta}$, from DB 1's viewpoint, the coefficient realization $\boldsymbol{x}$ is equally likely to be generated for $\boldsymbol{\theta}=(1,2),(2,1),(1,1)$ or $(2,2)$. Therefore, the scheme is private from DB 1's point of view. Due to symmetry, the scheme is also private from any other DB's viewpoint. As a result, the proposed delivery scheme is private.

Performance: Since $D=2N+2$ linear combinations, each containing one bit, are downloaded in total, the achieved load is $R=\frac{N+1}{N}$.

We provide the following example to illustrate the above design.

Example 1: (**Achievability of** $(1/4, 3/2)$ **for** $N=2$) Consider the cache-aided MuPIR problem with $K=K_{\mathrm{u}}=N=2$. The cache placement and private delivery phases are as follows.

3) Cache Placement: Each message consists of $L=4$ bits, i.e., $A=(A_1,A_2,A_3,A_4)$, $B=(B_1,B_2,B_3,B_4)$. Each user stores a linear combination of the message bits which are $Z_1=\boldsymbol{\alpha}_{1,1}[A_1,A_2]^{\mathsf{T}}+\boldsymbol{\beta}_{1,1}[B_1,B_2]^{\mathsf{T}}=A_1+B_1$, $Z_2=\boldsymbol{\alpha}_{2,1}[A_3,A_4]^{\mathsf{T}}+\boldsymbol{\beta}_{2,1}[B_3,B_4]^{\mathsf{T}}=A_3+B_3$ where the coefficients are chosen as $\boldsymbol{\alpha}_{1,1}=\boldsymbol{\alpha}_{2,1}=\boldsymbol{\beta}_{1,1}=\boldsymbol{\beta}_{2,1}=[1,0]$.[5] Therefore, $M=1/4$.

4) Private Delivery: The answers are

$$\begin{bmatrix} A_{1,1}^{[\boldsymbol{\theta}]} \\ A_{1,2}^{[\boldsymbol{\theta}]} \\ A_{2,1}^{[\boldsymbol{\theta}]} \\ A_{2,2}^{[\boldsymbol{\theta}]} \\ A_{2,3}^{[\boldsymbol{\theta}]} \\ A_{2,4}^{[\boldsymbol{\theta}]} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_{1,1} & \mathbf{0}_2 & \mathbf{v}_{1,1} & \mathbf{0}_2 \\ \mathbf{0}_2 & \mathbf{u}_{1,2} & \mathbf{0}_2 & \mathbf{v}_{1,2} \\ \mathbf{g}_1 & \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 \\ \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{g}_2 & \mathbf{0}_2 \\ \mathbf{0}_2 & \mathbf{g}_3 & \mathbf{0}_2 & \mathbf{0}_2 \\ \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{g}_4 \end{bmatrix} \begin{bmatrix} A_{(1:2)}^{\mathsf{T}} \\ A_{(3:4)}^{\mathsf{T}} \\ B_{(1:2)}^{\mathsf{T}} \\ B_{(3:4)}^{\mathsf{T}} \end{bmatrix}. \tag{24}$$

Suppose the demand vector is $(\theta_1,\theta_2)=(1,2)$. For this demand vector, we let $\mathbf{u}_{1,1}=\mathbf{g}_1$ and $\mathbf{v}_{1,2}=\mathbf{g}_4$ as shown in (12). To specify the coefficients in (24), we introduce the matrix $\mathbf{Y}_2=\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}$ which is independent of the $\boldsymbol{\theta}$. We let $\mathbf{g}_1$ and $\mathbf{g}_4$ be chosen randomly and uniformly i.i.d. from $\mathcal{Y}_2=\{[1,1],[0,1]\}$. In addition, we let $[\mathbf{u}_{1,2};\mathbf{g}_3]$ and $[\mathbf{v}_{1,1};\mathbf{g}_2]$ be two independent random permutations of the rows of $\mathbf{Y}_2$.

Correctness: From $A_{1,2}^{[\boldsymbol{\theta}]}-A_{2,4}^{[\boldsymbol{\theta}]}$, each user obtains $\mathbf{u}_{1,2}[A_3,A_4]^{\mathsf{T}}$. In addition, each user receives $A_{2,3}^{[\boldsymbol{\theta}]}=\mathbf{g}_3[A_3,A_4]^{\mathsf{T}}$. Since $\mathbf{u}_{1,2}$ and $\mathbf{g}_3$ are two different rows of $\mathbf{Y}_2$, it can be seen that they are linearly independent. Thus A_3 and A_4 can be decoded by the users. From $A_{1,1}^{[\boldsymbol{\theta}]}-A_{2,1}^{[\boldsymbol{\theta}]}$, each user obtains $\mathbf{v}_{1,1}[B_1,B_2]^{\mathsf{T}}$. Also, each user receives $A_{2,2}^{[\boldsymbol{\theta}]}=\mathbf{g}_2[B_1,B_2]^{\mathsf{T}}$. Since $\mathbf{v}_{1,1}$ and $\mathbf{g}_2$ are linearly independent, each user can decode B_1 and B_2. User 1 caches A_1+B_1 and has decoded B_1, it can then decode A_1. User 1 also receives $A_{2,1}^{[\boldsymbol{\theta}]}=\mathbf{g}_1[A_1,A_2]^{\mathsf{T}}$ where $\mathbf{g}_1\in\mathcal{Y}_2$. Due to the design of $\mathcal{Y}_2$, for both choices of $\mathbf{g}_1$, user 1 can always decode A_1 and A_2 from $\mathbf{g}_1[A_1,A_2]^{\mathsf{T}}$. Therefore, user 1 can recover message A. Similarly, since user 2 caches A_3+B_3 and has decoded A_3, it can then decode B_3. User 2 also receives $A_{2,4}^{[\boldsymbol{\theta}]}=\mathbf{g}_4[B_3,B_4]^{\mathsf{T}}$ where $\mathbf{g}_4\in\mathcal{Y}_2$. It can be seen that user 2 can always decode B_3,B_4 regardless of the choice of $\mathbf{g}_4$. Therefore, both users can recover their desired messages, proving the correctness of the scheme.

Privacy: From the viewpoint of DB 1, whose sent linear combinations are $\mathbf{g}_1[A_1,A_2]^{\mathsf{T}}+\mathbf{v}_{1,1}[B_1,B_2]^{\mathsf{T}}$ and $\mathbf{u}_{1,2}[A_3,A_4]^{\mathsf{T}}+\mathbf{g}_4[B_3,B_4]^{\mathsf{T}}$, the vectors $\mathbf{g}_1,\mathbf{g}_4,\mathbf{v}_{1,1},\mathbf{u}_{1,2}$ appear to be chosen randomly and uniformly i.i.d. from $\mathcal{Y}_2$ regardless of $\boldsymbol{\theta}$. Thus, these linear combinations are independent of $\boldsymbol{\theta}$. From the viewpoint of DB 2, whose sent linear combinations are $\mathbf{g}_1[A_1,A_2]^{\mathsf{T}}$, $\mathbf{g}_2[B_1,B_2]^{\mathsf{T}}$, $\mathbf{g}_3[A_3,A_4]^{\mathsf{T}}$, and $\mathbf{g}_4[B_3,B_4]^{\mathsf{T}}$, the vectors $\mathbf{g}_1,\mathbf{g}_2,\mathbf{g}_3,\mathbf{g}_4$ also appear to be chosen randomly and uniformly i.i.d. from $\mathcal{Y}_2$, implying the independence between the coefficients and $\boldsymbol{\theta}$. Therefore, each DB cannot get any information about $\boldsymbol{\theta}$ from its answer and the user cache.

Performance: The achieved load is $R=3/2$. ◊

B. Achievability of $\left(\frac{2(N-1)}{2N-1},\frac{N+1}{2N-1}\right)$

The cache placement and private delivery phases are described as follows.

1) Cache Placement: Let $W_1=A, W_2=B$ be the two messages each of which has $L=2N-1$ bits, i.e., $A=(A_1,\ldots,A_{2N-1})$, $B=(B_1,\ldots,B_{2N-1})$. Each user stores $2(N-1)$ bits of each message (therefore $M=\frac{2(N-1)}{2N-1}$), i.e.,

$$Z_1=\{A_{1:N-1},B_{1:N-1}\}, \tag{25a}$$

$$Z_2=\{A_{N:2N-2},B_{N:2N-2}\}. \tag{25b}$$

2) Private Delivery: We first construct the answers from the DBs. The answer of DB $n\in[N-1]$ is a linear combination of certain message bits which is $A_n^{[\boldsymbol{\theta}]}=\mathbf{u}_nA_{(1:2N-1)}^{\mathsf{T}}+\mathbf{v}_nB_{(1:2N-1)}^{\mathsf{T}}$. The answer of DB N consists of two linear combinations, i.e., $A_N^{[\boldsymbol{\theta}]}=\left(A_{N,1}^{[\boldsymbol{\theta}]},A_{N,2}^{[\boldsymbol{\theta}]}\right)$ where $A_{N,1}^{[\boldsymbol{\theta}]}=\mathbf{g}_1\ A_{(1:2N-1)}^{\mathsf{T}}$ and $A_{N,2}^{[\boldsymbol{\theta}]}=\mathbf{g}_2\ B_{(1:2N-1)}^{\mathsf{T}}$. The coefficient vectors $\mathbf{u}_n\triangleq[u_{n,1},\ldots,u_{n,2N-1}]$, $\mathbf{v}_n\triangleq[v_{n,1},\ldots,v_{n,2N-1}], \forall n\in[N-1]$, $\mathbf{g}_j\triangleq[g_{j,1},\ldots,g_{j,2N-1}], \forall j\in[2]$ belong to $\mathbb{F}_2^{1\times(2N-1)}\backslash\{\mathbf{0}_{2N-1}\}$ and are subject to design according to the user demands.

[5] With a slight abuse of notation, here we use Z_1, Z_2 to denote the cached bits by the users despite they are defined as sets.

These answers can be written as

$$\begin{bmatrix} A_1^{[\boldsymbol{\theta}]} \\ \vdots \\ A_{N-1}^{[\boldsymbol{\theta}]} \\ A_{N,1}^{[\boldsymbol{\theta}]} \\ A_{N,2}^{[\boldsymbol{\theta}]} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{v}_1 \\ \vdots & \vdots \\ \mathbf{u}_{N-1} & \mathbf{v}_{N-1} \\ \mathbf{g}_1 & \mathbf{0}_{2N-1} \\ \mathbf{0}_{2N-1} & \mathbf{g}_2 \end{bmatrix} \begin{bmatrix} A^{\mathsf{T}}_{(1:2N-1)} \\ B^{\mathsf{T}}_{(1:2N-1)} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{u}_{1,(1:2N-2)} & 1 & \mathbf{v}_{1,(1:2N-2)} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{u}_{N-1,(1:2N-2)} & 1 & \mathbf{v}_{N-1,(1:2N-2)} & 1 \\ \mathbf{g}_{1,(1:2N-2)} & 1 & \mathbf{0}_{2N-2} & 0 \\ \mathbf{0}_{2N-2} & 0 & \mathbf{g}_{2,(1:2N-2)} & 1 \end{bmatrix} \times \begin{bmatrix} A^{\mathsf{T}}_{(1:2N-1)} \\ B^{\mathsf{T}}_{(1:2N-1)} \end{bmatrix}, \qquad (26)$$

where we fix the coefficients $u_{n,2N-1} = v_{n,2N-1} = 1$, $\forall n \in [N-1]$ and $g_{1,2N-1} = g_{2,2N-1} = 1$.

We next consider different demands and present the necessary full-rank and alignment conditions. Due to space limit, we focus on the cases of $(\theta_1, \theta_2) = (1, 2)$ and $(1, 1)$.

For $(\theta_1, \theta_2) = (1, 2)$, the following two coefficient matrices are required to be full-rank:

Full-rank condition: the following matrices are full-rank,

$$\begin{bmatrix} \mathbf{u}_{1,(N:2N-1)} \\ \vdots \\ \mathbf{u}_{N-1,(N:2N-1)} \\ \mathbf{g}_{1,(N:2N-1)} \end{bmatrix}, \begin{bmatrix} [\mathbf{v}_{1,(1:N-1)}, v_{1,2N-1}] \\ \vdots \\ [\mathbf{v}_{N-1,(1:N-1)}, v_{N-1,2N-1}] \\ [\mathbf{g}_{2,(1:N-1)}, g_{2,2N-1}] \end{bmatrix}, \qquad (27)$$

where $\mathbf{u}_{1,(N:2N-1)} \triangleq [u_{1,N}, u_{1,N+1}, \ldots, u_{1,2N-1}]$ and other notation follows similarly. For alignment, we let $\forall n \in [N-1]$:

Alignment condition:

$$[\mathbf{u}_{n,(1:N-1)}, u_{n,2N-1}] = [\mathbf{g}_{1,(1:N-1)}, g_{1,2N-1}], \qquad (28a)$$

$$\mathbf{v}_{n,(N:2N-1)} = \mathbf{g}_{2,(N:2N-1)}, \qquad (28b)$$

i.e., the message bits $A_{(1:N-1)}, A_{2N-1}$ are aligned among the linear combinations $A_n^{[(1,2)]}, \forall n \in [N-1]$ and $A_{N,1}^{[(1,2)]}$; the bits $B_{(N:2N-1)}$ are aligned among $A_n^{[(1,2)]}, \forall n \in [N-1]$ and $A_{N,2}^{[(1,2)]}$. We next show that the users can recover their desired messages with the above conditions.

For user 1, due to the alignment of $B_{(N:2N-1)}$, we have

$$\begin{aligned} A_{N,2}^{[(1,2)]} - \mathbf{g}_{2,(1:N-1)} B^{\mathsf{T}}_{(1:N-1)} &= \mathbf{g}_{2,(N:2N-1)} B^{\mathsf{T}}_{(N:2N-1)} \\ &= \mathbf{v}_{n,(N:2N-1)} B^{\mathsf{T}}_{(N:2N-1)}, \\ &\quad \times \forall n \in [N-1]. \end{aligned} \qquad (29)$$

Subtracting $A_{N,2}^{[(1,2)]} - \mathbf{g}_{2,(1:N-1)} B^{\mathsf{T}}_{(1:N-1)}$ (this is known to user 1 since $B_{1:N-1}$ are already cached by user 1) from $A_1^{[(1,2)]}, \ldots, A_{N-1}^{[(1,2)]}$ in (26), together with $A_{N,1}^{[(1,2)]} = \mathbf{g}_1 A^{\mathsf{T}}_{(1:2N-1)}$, user 1 obtains N independent linear combinations of $A_{(N:2N-1)}$, which can be decoded as

$$A^{\mathsf{T}}_{(N:2N-1)} = \begin{bmatrix} \mathbf{u}_{1,(N:2N-1)} \\ \vdots \\ \mathbf{u}_{N-1,(N:2N-1)} \\ \mathbf{g}_{1,(N:2N-1)} \end{bmatrix}^{-1} \mathbf{y}, \qquad (30)$$

where $\mathbf{y}$ is defined as

$$\mathbf{y} \triangleq \begin{bmatrix} A_1^{[(1,2)]} - (A_{N,2}^{[(1,2)]} - \mathbf{g}_{2,(1:N-1)} B^{\mathsf{T}}_{(1:N-1)}) \\ \vdots \\ A_{N-1}^{[(1,2)]} - (A_{N,2}^{[(1,2)]} - \mathbf{g}_{2,(1:N-1)} B^{\mathsf{T}}_{(1:N-1)}) \\ A_{N,1}^{[(1,2)]} \end{bmatrix} - \begin{bmatrix} \mathbf{u}_{1,(1:N-1)} & \mathbf{v}_{1,(1:N-1)} \\ \vdots & \vdots \\ \mathbf{u}_{N-1,(1:N-1)} & \mathbf{v}_{N-1,(1:N-1)} \\ \mathbf{g}_{1,(1:N-1)} & \mathbf{0}_{N-1} \end{bmatrix} \begin{bmatrix} A^{\mathsf{T}}_{(1:N-1)} \\ B^{\mathsf{T}}_{(1:N-1)} \end{bmatrix}. \qquad (31)$$

Since the message bits $A_{(1:N-1)}, B_{(1:N-1)}$ are cached by user 1, it can decode the bits $A_{(N:2N-1)}$ and then recover message A. Similarly, user 2 can correctly recover message B.

For $(\theta_1, \theta_2) = (1, 1)$, the following two coefficient matrices should by full-rank:

$$\begin{bmatrix} \mathbf{u}_{1,(N:2N-1)} \\ \mathbf{u}_{2,(N:2N-1)} \\ \vdots \\ \mathbf{u}_{N-1,(N:2N-1)} \\ \mathbf{g}_{1,(N:2N-1)} \end{bmatrix}, \begin{bmatrix} [\mathbf{u}_{1,(1:N-1)}, u_{1,2N-1}] \\ [\mathbf{u}_{2,(1:N-1)}, u_{2,2N-1}] \\ \vdots \\ [\mathbf{u}_{N-1,(1:N-1)}, u_{N-1,2N-1}] \\ [\mathbf{g}_{1,(1:N-1)}, g_{1,2N-1}] \end{bmatrix}. \qquad (32)$$

The alignment condition is

$$\mathbf{g}_2 = \mathbf{v}_n, \quad \forall n \in [N-1]. \qquad (33)$$

The decoding process is explained as follows. Due to the alignment of (33), we have

$$A_{N,2}^{[(1,1)]} = \mathbf{g}_2 B^{\mathsf{T}}_{(1:2N-1)} = \mathbf{v}_n B^{\mathsf{T}}_{(1:2N-1)}, \quad \forall n \in [N-1]. \qquad (34)$$

Subtracting $A_{N,2}^{[(1,1)]}$ from all $A_n^{[(1,1)]}, \forall n \in [N-1]$, we obtain

$$\begin{aligned} &A^{\mathsf{T}}_{(N:2N-1)} \\ &= \begin{bmatrix} \mathbf{u}_{1,(N:2N-1)} \\ \mathbf{u}_{2,(N:2N-1)} \\ \vdots \\ \mathbf{u}_{N-1,(N:2N-1)} \\ \mathbf{g}_{1,(N:2N-1)} \end{bmatrix}^{-1} \\ &\times \left(\begin{bmatrix} A_1^{[(1,1)]} - A_{N,2}^{[(1,1)]} \\ A_2^{[(1,1)]} - A_{N,2}^{[(1,1)]} \\ \vdots \\ A_{N-1}^{[(1,1)]} - A_{N,2}^{[(1,1)]} \\ A_{N,1}^{[(1,1)]} \end{bmatrix} - \begin{bmatrix} \mathbf{u}_{1,(1:N-1)} \\ \mathbf{u}_{2,(1:N-1)} \\ \vdots \\ \mathbf{u}_{N-1,(1:N-1)} \\ \mathbf{g}_{1,(1:N-1)} \end{bmatrix} A^{\mathsf{T}}_{(1:N-1)} \right). \end{aligned} \tag{35}$$

Since the bits $A_{1:N-1}$ are cached by user 1, it can decode the desired bits $A_{N:2N-1}$ and then recover A. Similarly, user 2 can decode the desired bits $A_{1:N-1}, A_{2N-1}$. Therefore, both users can correctly recover message A.

With the above full-rank and alignment conditions, we now employ a randomized specification of the linear coefficients used by each DB and formally describe the delivery scheme.

We first introduce some necessary notation. Let $\mathbf{Y}'_N \triangleq [\mathbf{I}_{N-1}; \mathbf{0}_{N-1}] \in \mathbb{F}_2^{N\times(N-1)}$ and let $\mathcal{Y}'_N \triangleq \{\mathbf{Y}'_N(i,:) : i \in [N]\}$ be a set containing the rows of $\mathbf{Y}'_N$. For an index vector $(m:n) = (m, m+1, \ldots, n)$, define $\forall j \in [2]$:

$$\mathbf{M}'(\mathbf{u}, \mathbf{g}_j, (m:n)) \triangleq [\mathbf{u}_{1,(m:n)}; \ldots; \mathbf{u}_{N-1,(m:n)}; \mathbf{g}_{j,(m:n)}] \in \mathbb{F}_2^{N\times(n-m+1)}, \tag{36a}$$

$$\mathbf{M}'(\mathbf{v}, \mathbf{g}_j, (m:n)) \triangleq [\mathbf{v}_{1,(m:n)}; \ldots; \mathbf{v}_{N-1,(m:n)}; \mathbf{g}_{j,(m:n)}] \in \mathbb{F}_2^{N\times(n-m+1)}. \tag{36b}$$

It can be seen that $\mathbf{M}'(\mathbf{u}, \mathbf{g}_j, (m:n))$ and $\mathbf{M}'(\mathbf{v}, \mathbf{g}_j, (m:n))$ represent the coefficient sub-matrices of (26) corresponding to the message bits $A_{(m:n)}$ and $B_{(m:n)}$ respectively.

The delivery strategies for different demands are as follows.

- $(\theta_1, \theta_2) = (1,2)$: Let $\mathbf{g}_{1,(1:N-1)}$ and $\mathbf{g}_{2,(N:2N-2)}$ be chosen randomly and uniformly i.i.d. from $\mathcal{Y}'_N$. Also, let $\mathbf{M}'(\mathbf{u}, \mathbf{g}_1, (N:2N-2))$ and $\mathbf{M}'(\mathbf{v}, \mathbf{g}_2, (1:N-1))$ be two independent random permutations of the rows of $\mathbf{Y}'_N$. It can be easily seen that the full-rank condition of (27) is satisfied, guaranteeing the decodability.
- $(\theta_1, \theta_2) = (1,1)$: Let $\mathbf{g}_{2,(N:2N-2)}$ and $\mathbf{g}_{2,(1:N-1)}$ be chosen randomly and uniformly i.i.d. from $\mathcal{Y}'_N$. Also, let $\mathbf{M}'(\mathbf{u}, \mathbf{g}_1, (N:2N-2))$ and $\mathbf{M}'(\mathbf{u}, \mathbf{g}_1, (1:N-1))$ be chosen as two independent random permutations of the rows of $\mathbf{Y}'_N$.
- $(\theta_1, \theta_2) = (2,1)$: Let $\mathbf{g}_{1,(N:2N-2)}$ and $\mathbf{g}_{2,(1:N-1)}$ be chosen randomly and uniformly i.i.d. from $\mathcal{Y}'_N$. Also, let $\mathbf{M}'(\mathbf{u}, \mathbf{g}_1, (1:N-1))$ and $\mathbf{M}'(\mathbf{v}, \mathbf{g}_2, (N:2N-2))$ be chosen as two independent random permutations of the rows of $\mathbf{Y}'_N$.
- $(\theta_1, \theta_2) = (2,2)$: Let $\mathbf{g}_{1,(N:2N-2)}$ and $\mathbf{g}_{1,(1:N-1)}$ be chosen randomly and uniformly i.i.d. from $\mathcal{Y}'_N$. Also, let $\mathbf{M}'(\mathbf{v}, \mathbf{g}_2, (1:N-1))$ and $\mathbf{M}'(\mathbf{v}, \mathbf{g}_2, (N:2N-2))$ be chosen as two independent random permutations of the rows of $\mathbf{Y}'_N$.

Correctness: Decodability is straightforward since the randomized specifications of the linear coefficients guarantee the corresponding full-rank and alignment conditions.

Privacy: The privacy can be proved by a similar argument to that of (23).

Performance: Since $D = N+1$ linear combinations, each containing one bit, are downloaded in total, the achieved load is $R = \frac{N+1}{2N-1}$.

The following example is provided to illustrate the above design idea.

Example 2: (**Achievability of** $(2/3, 1)$ **for** $N = 2$) Consider the same setting as Example 1 where $K = K_{\mathrm{u}} = N = 2$. We show the achievability of the memory-load pair $(2/3, 1)$ which is achieved by uncoded placement.

3) Cache Placement: Each message consists of $L = 3$ bits, i.e., $A = (A_1, A_2, A_3)$, $B = (B_1, B_2, B_3)$. The cache placement is $Z_1 = \{A_1, B_1\}, Z_2 = \{A_2, B_2\}$. Therefore, $M = 2/3$.

4) Private Delivery: The answers are written as

$$\begin{aligned} \begin{bmatrix} A_1^{[\boldsymbol{\theta}]} \\ A_{2,1}^{[\boldsymbol{\theta}]} \\ A_{2,2}^{[\boldsymbol{\theta}]} \end{bmatrix} &= \begin{bmatrix} \mathbf{u}_1 & \mathbf{v}_1 \\ \mathbf{g}_1 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{g}_2 \end{bmatrix} \begin{bmatrix} A^{\mathsf{T}}_{(1:3)} \\ B^{\mathsf{T}}_{(1:3)} \end{bmatrix} \\ &= \begin{bmatrix} u_{1,1} & u_{1,2} & 1 & v_{1,1} & v_{1,2} & 1 \\ g_{1,1} & g_{1,2} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & g_{2,1} & g_{2,2} & 1 \end{bmatrix} \begin{bmatrix} A^{\mathsf{T}}_{(1:3)} \\ B^{\mathsf{T}}_{(1:3)} \end{bmatrix}. \end{aligned} \tag{37}$$

Suppose $(\theta_1, \theta_2) = (1,2)$. For this demand vector, we let $u_{1,1} = g_{1,1}$, $v_{1,2} = g_{2,2}$ as shown in (28). Thus (37) becomes

$$\begin{bmatrix} A_1^{[\boldsymbol{\theta}]} \\ A_{2,1}^{[\boldsymbol{\theta}]} \\ A_{2,2}^{[\boldsymbol{\theta}]} \end{bmatrix} = \begin{bmatrix} g_{1,1} & u_{1,2} & 1 & v_{1,1} & g_{2,2} & 1 \\ g_{1,1} & g_{1,2} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & g_{2,1} & g_{2,2} & 1 \end{bmatrix} \begin{bmatrix} A^{\mathsf{T}}_{(1:3)} \\ B^{\mathsf{T}}_{(1:3)} \end{bmatrix}. \tag{38}$$

To specify the coefficients, we let $g_{1,1}$ and $g_{2,2}$ be chosen randomly and uniformly i.i.d. from $\{0,1\}$. Also, let $[u_{1,2}, g_{1,2}]^{\mathsf{T}}$ and $[v_{1,1}, g_{2,1}]^{\mathsf{T}}$ be two independent random permutations of $[1,0]^{\mathsf{T}}$.

Correctness: We first consider user 1 who stores A_1, B_1 and wants message A. From $A_{2,1}^{[\boldsymbol{\theta}]}$ and A_1, user 1 obtains $g_{1,2}A_2 + A_3$. From $A_1^{[\boldsymbol{\theta}]} - A_{2,2}^{[\boldsymbol{\theta}]}$, user 1 decodes $g_{1,1}A_1 + u_{1,2}A_2 + A_3 + (v_{1,1} - g_{2,1})B_1$, and then obtains $u_{1,2}A_2 + A_3$ by removing the cached bits A_1, B_1. Since $u_{1,2} \neq g_{1,2}$, user 1 can solve A_2, A_3 from the two independent linearly combinations $g_{1,2}A_2 + A_3$ and $u_{1,2}A_2 + A_3$. Therefore, user 1 recovers message A. We then consider user 2 who stores A_2, B_2 and wants message B. From $A_{2,2}^{[\boldsymbol{\theta}]}$ and B_2, user 2 obtains $g_{2,1}B_1 + B_3$. From $A_1^{[\boldsymbol{\theta}]} - A_{2,1}^{[\boldsymbol{\theta}]}$ and the cached bits A_2, B_2, user 2 obtains $v_{1,1}B_1 + B_3$. Since $v_{1,1} \neq g_{2,1}$, the two linear combinations $g_{2,1}B_1 + B_3$ and $v_{1,1}B_1 + B_3$ are independent, from which B_1, B_3 can be solved by user 2. Therefore, both users can recover their desired messages.

Privacy: From the viewpoint of DB 1 whose sent linear combination is $g_{1,1}A_1 + u_{1,2}A_2 + A_3 + v_{1,1}B_1 + g_{2,2}B_2 + B_3$, the coefficients $g_{1,1}, u_{1,2}, v_{1,1}, g_{2,2}$ appear to be chosen randomly and uniformly i.i.d. from $\{0, 1\}$ regardless of $\boldsymbol{\theta}$. Thus the answer of DB 1 is independent of $\boldsymbol{\theta}$. From the viewpoint of DB 2 whose sent linear combinations are $g_{1,1}A_1 + g_{1,2}A_2 + A_3$ and $g_{2,1}B_1 + g_{2,2}B_2 + B_3$, the coefficients $g_{1,1}, g_{1,2}, g_{2,1}, g_{2,2}$ also appear to be chosen randomly and uniformly i.i.d. from $\{0, 1\}$, implying the independence between the DB 2 answer and $\boldsymbol{\theta}$. Therefore, the delivery scheme is private from both users' viewpoint.

Performance: The achieved load is $R = 1$ for $M = 2/3$. $\diamondsuit$

V. Proof of Theorem 2: The Product Design

In this section, we present a general achievable scheme for arbitrary $K, K_{\rm u}$ and N, which we call *Product Design* (PD). PD is inspired by both MAN coded caching [6] and the SJ PIR scheme [5] and enjoys combined coding gain.[6] By comparing with the already established converse bounds for coded caching [33], we show that the PD is optimal within a factor 8 in general as implied by Theorem 2. We first provide an example to highlight the design idea of PD and then present the general achievable schemes.

A. An Example

Example 3: Consider the cache-aided MuPIR problem with $K = 3$ messages, $K_{\rm u} = 2$ users and $N = 2$ DBs. Let A, B and C denote the three messages. By Theorem 2, the pair $(1, 7/8)$ is achievable.

1) Cache Placement: Each message is split into two *packets*, i.e., $A = (A_1, A_2)$ where A_i represents the i-the packet which consists of 8 bits. Therefore, $L = 16$ bits. Similarly, $B = (B_1, B_2)$, $C = (C_1, C_2)$. User $u \in [2]$ then stores one packet of each message, i.e., the cache is $Z_u = \{A_u, B_u, C_u\}$, satisfying the memory constraint $M = 3/2$.

2) Private Delivery: Let $[A_i^1, \ldots, A_i^8]$, $[B_i^1, \ldots, B_i^8]$ and $[C_i^1, \ldots, C_i^8]$ represent three independent random permutations of the 8 bits of the packet A_i, B_i and $C_i, \forall i \in [2]$ respectively. These permutations are known to the users but not to the DBs. Suppose $\boldsymbol{\theta} = (1, 2)$. Let $A_n^{[\theta_1]}(A_2, B_2, C_2)$ represent the answer of DB $n \in [2]$ in the SJ PIR scheme where the corresponding message library is (First messages, second message, third messages) $= (A_2, B_2, C_2)$ and the user demand is θ_1 (i.e., the user demands A_2). Other notations follow similarly. Then the answer of DB n is constructed as $A_n^{[\boldsymbol{\theta}]} = A_n^{[\theta_1]}(A_2, B_2, C_2) + A_n^{[\theta_2]}(A_1, B_1, C_1), \forall n \in [2]$, which is shown in Table I. For other demands, the delivery phase proceeds similarly. We next prove that the above delivery scheme is both correct and private.

TABLE I

Answer of Example 3

DB 1 ($A_1^{[\boldsymbol{\theta}]}$)	DB 2 ($A_2^{[\boldsymbol{\theta}]}$)
$A_2^1 + A_1^1$	$A_2^2 + A_1^2$
$B_2^1 + B_1^1$	$B_2^2 + B_1^2$
$C_2^1 + C_1^1$	$C_2^2 + C_1^2$
$A_2^3 + B_2^2 + A_1^2 + B_1^3$	$A_2^5 + B_2^1 + A_1^3 + B_1^5$
$A_2^4 + C_2^2 + A_1^2 + C_1^3$	$A_2^6 + C_2^1 + A_1^4 + C_1^4$
$B_2^3 + C_2^2 + B_1^4 + C_1^2$	$B_2^4 + C_2^4 + B_1^6 + C_1^1$
$A_2^7 + B_2^4 + C_2^4 + A_1^4 + B_1^7 + C_1^4$	$A_2^8 + B_2^3 + C_2^3 + A_1^2 + B_1^8 + C_1^3$

Correctness: For $\boldsymbol{\theta} = (1, 2)$, user 1 needs A_2 since A_1 is already in the cache. Similarly, user 2 needs B_1. For user 1, since A_1, B_1 and C_1 are stored in Z_1, the interference of $A_n^{[\theta_2]}(A_1, B_1, C_1)$ can be eliminated from $A_n^{[\boldsymbol{\theta}]}, \forall n \in [2]$. Then user 1 obtains $A_1^{[\theta_1]}(A_2, B_2, C_2)$ and $A_2^{[\theta_1]}(A_2, B_2, C_2)$, from which A_2 can be decoded as in the SJ scheme. Similarly, by eliminating A_2, B_2 and C_2 from the answers, user 2 can decode B_1. Therefore, both users can recover their desired messages.

Privacy: We show that the delivery scheme is private from DB 1's viewpoint. First, by the privacy of the SJ scheme, DB 1 can neither determine which of the packets A_2, B_2 and C_2 is requested by user 1, nor determine which of the packets A_1, B_1 and C_1 is requested by user 2. Second, note that random permutations that are applied to the set of packets $\{A_1, B_1, C_1\}$ and $\{Z_2, B_2, C_2\}$ are independent. These two aspects guarantee the privacy of $\boldsymbol{\theta}$ with respect to DB 1. By symmetry, the scheme is also private from DB 1's viewpoint.

Performance: Since $D = 14$ bits are downloaded in total, the achieved load is $R = 7/8$. $\diamondsuit$

B. General Achievable Scheme

For general $K, K_{\rm u}$ and N, we assume that $t = \frac{K_{\rm u}M}{K} \in [1 : K_{\rm u}]$. Each message is assumed to have $L = \binom{K_{\rm u}}{t}N^K$ bits. The cache placement and delivery phases are described as follows.

1) Cache Placement: The MAN cache placement is applied over the message packets. In particular, each message W_k is split into $\binom{K_{\rm u}}{t}$ disjoint and equal-sized packets, i.e., $W_k \overset{\Delta}{=} \{W_{k,\mathcal{T}} : \mathcal{T} \subseteq [K_{\rm u}], |\mathcal{T}| = t\}, \forall k \in [K]$. Therefore, each packet consists of $\frac{L}{\binom{K_{\rm u}}{t}} = N^K$ bits. User u then stores all the packets $W_{k,\mathcal{T}}$ such that $u \in \mathcal{T}$, i.e.,

$$Z_u = \{W_{k,\mathcal{T}} : \mathcal{T} \subseteq [K_{\rm u}], |\mathcal{T}| = t, u \in \mathcal{T}, \forall k \in [K]\}, \quad (39)$$

for each $u \in [K_{\rm u}]$. Therefore, each user stores $KL\frac{\binom{K_{\rm u}-1}{t-1}}{\binom{K_{\rm u}}{t}} = ML$ bits, satisfying the memory size constraint.

2) Private Delivery: Suppose the user demands are $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{K_{\rm u}})$. We first construct $\binom{K_{\rm u}}{t+1}$ different coded messages

$$\left\{X_{\mathcal{S}}^{[\boldsymbol{\theta}]} \overset{\Delta}{=} \left(A_{1,\mathcal{S}}^{[\boldsymbol{\theta}]}, A_{2,\mathcal{S}}^{[\boldsymbol{\theta}]}, \ldots, A_{N,\mathcal{S}}^{[\boldsymbol{\theta}]}\right) : \mathcal{S} \subseteq [K_{\rm u}], |\mathcal{S}| = t + 1\right\}, \quad (40)$$

each of which being useful to a subset of $t+1$ users in $\mathcal{S}$. The n-th component $A_{n,\mathcal{S}}^{[\boldsymbol{\theta}]}$ of $X_{\mathcal{S}}^{[\boldsymbol{\theta}]}$ represents the answer from DB n. For each $\mathcal{S}$, the components of the coded message $X_{\mathcal{S}}^{[\boldsymbol{\theta}]}$ are constructed as

$$A_{n,\mathcal{S}}^{[\boldsymbol{\theta}]} = \sum_{u \in \mathcal{S}} A_n^{[\theta_u]}\left(W_{1:K,\mathcal{S}\setminus\{u\}}\right), \quad \forall n \in [N], \quad (41)$$

[6] Note that, any capacity-achieving PIR scheme for the original single-user non-colluding database PIR problem can be used to combine with the linear caching code to produce a corresponding product design.

where $W_{1:K,\mathcal{S}\backslash\{u\}} \overset{\Delta}{=} \left(W_{1,\mathcal{S}\backslash\{u\}},\ldots,W_{K,\mathcal{S}\backslash\{u\}}\right)$. The term $A_n^{[\theta_u]}(W_{1:K,\mathcal{S}\backslash\{u\}})$ in the summation of (41) represents the answer from DB n in the SJ scheme for the single-user PIR problem when the messages are (First message, second message, …, K-th message) $= \left(W_{1,\mathcal{S}\backslash\{u\}},\ldots,W_{K,\mathcal{S}\backslash\{u\}}\right)$ and user u demands $W_{\theta_u,\mathcal{S}\backslash\{u\}}$. To preserve demand privacy, the users employ a random and uniform i.i.d. permutation (not known to the DBs) of the N^K bits of each packet $W_{\theta_u,\mathcal{S}\backslash\{u\}}$. For different coded messages $X_{\mathcal{S}}^{[\boldsymbol{\theta}]}$, the set of the random bit permutations of the corresponding packets are also independent from each other. Moreover, the ordering of the randomly-permuted message bits in the query to the DBs is preserved as in the SJ scheme in the summation of (41). Next we prove that the proposed PD is both correct and private.

Correctness: We show that for any user $u \in [K_{\mathrm{u}}]$, it can correctly recover its desired message W_{θ_u} from all the answers received. Since the packets $\{W_{k,\mathcal{T}} : |\mathcal{T}| = t, u \in \mathcal{T}, \forall k \in [K]\}$ are already cached by user u, it needs to recover the packets $\{W_{\theta_u,\mathcal{T}} : |\mathcal{T}| = t, u \notin \mathcal{T}\}$. For each $n \in [N], \mathcal{S} \subseteq [K_{\mathrm{u}}]$ such that $|\mathcal{S}| = t+1, u \in \mathcal{S}$, we can write (41) as

$$A_{n,\mathcal{S}}^{[\boldsymbol{\theta}]} = A_n^{[\theta_u]}\left(W_{1:K,\mathcal{S}\backslash\{u\}}\right) + \sum_{u' \in \mathcal{S}\backslash\{u\}} A_n^{[\theta_{u'}]}\left(W_{1:K,\mathcal{S}\backslash\{u'\}}\right), \tag{42}$$

from which user u can decode the desired term $A_n^{[\theta_u]}\left(W_{1:K,\mathcal{S}\backslash\{u\}}\right)$ since all the packets $\{W_{1:K,\mathcal{S}\backslash\{u'\}} : u' \neq u\}$ are cached by user u because $u \in \mathcal{S}\backslash\{u'\}$. Therefore, user u obtains a set of desired answers $\left\{A_n^{[\theta_u]}\left(W_{1:K,\mathcal{S}\backslash\{u\}}\right) : \forall n \in [N]\right\}$ from which the desire packet $W_{\theta_u,\mathcal{S}\backslash\{u\}}$ can be decoded due to the decodability of the SJ scheme. Going through all different $\mathcal{S}$, user u can decode all the $\binom{K_{\mathrm{u}}-1}{t}$ desired packets. As a result, user u can correctly recover its desired message W_{θ_u}.

Privacy: It can be seen that each $A_{n,\mathcal{S}}^{[\boldsymbol{\theta}]}, \forall n \in [N]$ of the coded message $X_{\mathcal{S}}^{[\boldsymbol{\theta}]}$ is independent of the demands of the users in $\mathcal{S}$ and that of the users in $[K_{\mathrm{u}}]\backslash\mathcal{S}$ from the viewpoint of each DB. The reason is explained as follows. For any user $u \in \mathcal{S}$, the first term in (42), i.e., $A_n^{[\theta_u]}\left(W_{1:K,\mathcal{S}\backslash\{u\}}\right), \forall n \in [N]$ is independent of θ_u by the privacy of the SJ scheme. Also, each term in the summation in (42) is independent of θ_u because for each $A_n^{[\theta_{u'}]}\left(W_{1:K,\mathcal{S}\backslash\{u'\}}\right)$, a set of random and independent permutations are employed to the bits of the set of packets $\{W_{k,\mathcal{S}\backslash\{u'\}} : \forall k \in [K]\}$. Therefore, $A_{n,\mathcal{S}}^{[\boldsymbol{\theta}]}$ is independent of the demands of the users in $\mathcal{S}$. Moreover, for $\mathcal{S}$ where $u \notin \mathcal{S}$, due to the employment of the random and independent permutations, $A_{n,\mathcal{S}}^{[\boldsymbol{\theta}]}$ is independent of the demands of the users in $[K_{\mathrm{u}}]\backslash\mathcal{S}$. As a result, $A_{n,\mathcal{S}}^{[\boldsymbol{\theta}]}$ is independent of $\boldsymbol{\theta}$ from DB n's viewpoint for any $\mathcal{S} \subseteq [K_{\mathrm{u}}], |\mathcal{S}| = t+1$, which completes the proof of privacy.

Performance: By the SJ PIR scheme, each $X_{\mathcal{S}}^{[\boldsymbol{\theta}]}$ has $\left(1 + \cdots + \frac{1}{N^{K-1}}\right) N^K$ bits. Therefore, $D = \binom{K_{\mathrm{u}}}{t+1}\left(1 + \cdots + \frac{1}{N^{K-1}}\right) N^K$. As a result, the achieved load is $\widehat{R}(M) = \frac{K_{\mathrm{u}}-t}{t+1}\left(1 + \cdots + \frac{1}{N^{K-1}}\right)$.

VI. Discussion: MuPIR With Distinct Demands

In this section, we consider an interesting scenario where the users have distinct demands. Recall that $R_{\mathrm{d}}^{\star}$ denotes the minimum load. We obtain the following theorem.

Theorem 3: For the cache-aided MuPIR problem with $K = 2$ messages, $K_{\mathrm{u}} = 2$ users and $N = 2$ DBs, where the users demand distinct messages in a uniform manner, the optimal memory-load trade-off is characterized as

$$R_{\mathrm{d}}^{\star}(M) = \begin{cases} 2(1-M), & 0 \le M \le 1/3 \\ 5/3 - M, & 1/3 \le M \le 2/3 \\ \frac{3(2-M)}{4}, & 2/3 \le M \le 2 \end{cases} \tag{43}$$

Proof: For achievability, we show that the memory-load pairs $(1/3, 4/3)$ and $(2/3, 1)$ are achievable using the idea of CIA in Section VI-A. Together with the two trivial pairs $(0, 2)$ and $(2, 0)$, we obtain four corner points. By the memory-sharing among these corner points, the load of Theorem 3 can be achieved. For the converse, when $M \le 1/3$, the load of $R_{\mathrm{d}}(M) = 2(1-M)$ coincides with the caching bound without demand privacy and hence is optimal. When $M \ge 2/3$, the load $R_{\mathrm{d}}(M) = \frac{3(2-M)}{4}$ is optimal since it coincides with the single-user cache-aided PIR bound of [9]. A novel converse bound is derived for the case $1/3 \le M \le 2/3$ to show the optimality of $R(M) = 5/3 - M$ when the users have distinct demands. ■

Remark 3: It can be seen that the load in (43) is lower than the one in (1) achieved by the CIA based scheme. Thus the optimal load under the constraint of distinct demands can be strictly lower than the optimal load without such constraint (See Fig. 2a). The reason is that, by removing the cases of identical demands, the decodability constraint is relaxed and therefore allows a lower achievable load.

A. Achievability

First, we consider the achievability of the memory-load pair $(1/3, 4/3)$. Assume that the users have distinct demands, i.e., $\boldsymbol{\theta} = (\theta_1, \theta_2) \in \{(1,2), (2,1)\}$.

1) ***Cache placement:*** Assume that each message contains $L = 3$ bits, i.e., $W_1 = (a_1, a_2, a_3)$, $W_2 = (b_1, b_2, b_3)$. The cache placement is $Z_1 = \{a_1 + b_1\}$, $Z_2 = \{a_2 + b_2\}$ and therefore $M = 1/3$.

2) ***Private delivery:*** Let $A_{1,1}$ and $A_{1,2}$ be two different answers of DB 1, and let $A_{2,1}$ and $A_{2,2}$ be two different answers of DB 2. The answers are

$$\begin{aligned} A_{1,1} &= (a_3, b_1 + b_2 + b_3), \\ A_{1,2} &= (a_1 + a_2 + a_3, b_3), \\ A_{2,1} &= (a_2 + a_3, b_2 + b_3), \\ A_{2,2} &= (a_1 + a_3, b_1 + b_3). \end{aligned} \tag{44}$$

The delivery scheme is that the users randomly choose $A_{1,1}$ or $A_{1,2}$ to request from DB 1 with equal probabilities. We then consider the following two cases. When $(\theta_1, \theta_2) = (1, 2)$, if $A_{1,1}$ is chosen, then go to DB 2 to download $A_{2,1}$. Otherwise, if $A_{1,2}$ is chosen, go to DB 2 to download $A_{2,2}$. When $(\theta_1, \theta_2) = (2, 1)$, if $A_{1,1}$ is chosen, then go to DB 2 to download $A_{2,2}$. Otherwise if $A_{1,2}$ is chosen, go to DB 2 to download $A_{2,1}$.

Correctness: One can check that $(A_{1,1}, A_{2,1}, Z_1) \rightarrow W_1$ (meaning that W_1 can be recovered from $A_{1,1}, A_{2,1}$ and Z_1), $(A_{1,1}, A_{2,1}, Z_2) \rightarrow W_2$, $(A_{1,2}, A_{2,2}, Z_1) \rightarrow W_1$, $(A_{1,2}, A_{2,2}, Z_2) \rightarrow W_2$, $(A_{1,1}, A_{2,2}, Z_1) \rightarrow W_2$, $(A_{1,1}, A_{2,2}, Z_2) \rightarrow W_1$, $(A_{1,2}, A_{2,1}, Z_1) \rightarrow W_2$, and $(A_{1,2}, A_{2,1}, Z_2) \rightarrow W_1$. Therefore, both users can decode their desired messages.

Privacy: Note that the answer from DB 1 is equally likely to be $A_{1,1}$ or $A_{1,2}$, and the answer from DB 2 is also equally likely to be $A_{2,1}$ or $A_{2,2}$. Therefore, we have

$$P(\boldsymbol{\theta} = (1,2)) = P((A_{1,1}, A_{2,1})) + P((A_{1,2}, A_{2,2})) = 1/2, \tag{45a}$$

$$P(\boldsymbol{\theta} = (2,1)) = P((A_{1,1}, A_{2,2})) + P((A_{1,2}, A_{2,1})) = 1/2. \tag{45b}$$

such that the privacy constraint (3) is satisfied (for distinct demands). Since $D = 4$ bits are downloaded, the achieved load is $R_{\text{d}} = 4/3$.

Second, we consider the achievability of $(2/3, 1)$. Let $W_1 = (a_1, a_2, a_3), W_2 = (b_1, b_2, b_3)$. The cache placement is $Z_1 = \{a_1, b_1\}, Z_2 = \{a_2, b_2\}$ and the answers are

$$\begin{aligned} A_{1,1} &= (a_3 + b_3 + b_1 + b_2), \\ A_{1,2} &= (a_3 + b_3 + a_1 + a_2), \\ A_{2,1} &= (a_2 + a_3,\ b_2 + b_3), \\ A_{2,2} &= (a_1 + a_3,\ b_1 + b_3). \end{aligned} \tag{46}$$

The delivery scheme is similarly to that of $(1/3, 4/3)$. The correctness of this scheme can be easily verified. The privacy argument is similar to the previous case, i.e., from each DB's viewpoint, the demand vector $\boldsymbol{\theta}$ is equally likely to be $(1,2)$ or $(2,1)$. Since $D = 3$ bits are downloaded in total, the achieved load is $R_{\text{d}} = 1$.

B. Converse

The converse consists of three piece-wise linear segments corresponding to different cache memory regimes $M \in [0, 1/3]$, $[1/3, 2/3]$ and $[2/3, 2]$. We prove the converse for each segment respectively. When $M \in [0, 1/3]$, the cut-set bound $R \geq 2(1-M)$ without privacy constraint is tight. Since increasing the number of users while preserving the demand privacy can only possibly increase the load, the single-user cache-aided PIR converse given in [9] is also a converse for our considered MuPIR problem. This gives a bound $R_{\text{d}}^{\star}(M) \geq \frac{3(2-M)}{4}$ for $M \in [2/3, 2]$. When $M \in [1/3, 2/3]$, we develop a new converse $R_{\text{d}}(M) \geq 5/3 - M$ as follows.

Let $A_{1,1} = A_{1,1}^{[(1,2)]} = A_{1,1}^{[(2,1)]}$ be an answer of DB 1 and let $A_{2,1} = A_{2,1}^{[(1,2)]} = A_{2,1}^{[(2,1)]}$ be an answer of DB 2. It is clear that the message W_1 can be recovered from $\{A_{1,1}^{[(1,2)]}, A_{2,1}^{[(1,2)]}, Z_1\}$ while W_2 can be recovered from $\{A_{1,1}^{[(1,2)]}, A_{2,1}^{[(1,2)]}, Z_2\}$ for which we use a shorthand notation as $\big(A_{1,1}^{[(1,2)]}, A_{2,1}^{[(1,2)]}, Z_1\big) \rightarrow W_1$, $\big(A_{1,1}^{[(1,2)]}, A_{2,1}^{[(1,2)]}, Z_2\big) \rightarrow W_2$. For privacy with respect to DB 1, there must exist another answer $A_{2,2}^{[(2,1)]}$ of DB 2 such that the demands $\boldsymbol{\theta} = (2,1)$ can be satisfied, i.e., $\big(A_{1,1}^{[(2,1)]}, A_{2,2}^{[(2,1)]}, Z_1\big) \rightarrow W_2$ and $\big(A_{1,1}^{[(2,1)]}, A_{2,2}^{[(2,1)]}, Z_2\big) \rightarrow W_1$. Also, for privacy with respect to DB 2, there must exist another answer $A_{1,2}^{[(2,1)]}$ of DB 1 such that $\big(A_{1,2}^{[(2,1)]}, A_{2,1}^{[(2,1)]}, Z_1\big) \rightarrow W_2$ and $\big(A_{1,2}^{[(2,1)]}, A_{2,1}^{[(2,1)]}, Z_2\big) \rightarrow W_1$. Note that $R_{\text{d}} = \frac{H(A_{1,i}) + H(A_{2,j})}{L}$ for any index pair $(i,j) \in \{(1,1), (1,2), (2,1)\}$ because the load does not depend on the demands. Denote $X_{i,j,k}^{[\boldsymbol{\theta}]} \triangleq \big(A_{1,i}^{[\boldsymbol{\theta}]}, A_{2,j}^{[\boldsymbol{\theta}]}, Z_k\big)$, $\forall (i,j,k) \in \{(1,1,1), (1,2,2), (2,1,2)\}$. Then

$$\begin{aligned} & 3R_{\text{d}}(M)L + 3ML \\ & \quad \geq H\big(X_{1,1,1}^{[(1,2)]}\big) + H\big(X_{2,1,2}^{[(2,1)]}\big) + H\big(X_{1,2,2}^{[(2,1)]}\big) \\ & \quad \overset{\text{(a)}}{=} 3L + H\big(X_{1,1,1}^{[(1,2)]}\big|W_1\big) + H\big(X_{2,1,2}^{[(2,1)]}\big|W_1\big) \\ & \qquad + H\big(X_{1,2,2}^{[(2,1)]}\big|W_1\big) \\ & \quad \overset{\text{(b)}}{\geq} 3L + H\big(X_{1,1,1}^{[(1,2)]}\big|W_1\big) + H(Z_2|W_1) \\ & \qquad + H\big(A_{2,1}^{[(2,1)]}\big|W_1, Z_2\big) \\ & \qquad + H\big(X_{1,2,2}^{[(2,1)]}\big|W_1\big) \\ & \quad \geq 3L + H\big(X_{1,1,1}^{[(1,2)]}, Z_2\big|W_1\big) + H\big(A_{2,1}^{[(2,1)]}\big|W_1, Z_2\big) \\ & \qquad + H\big(X_{1,2,2}^{[(2,1)]}\big|W_1\big) \\ & \quad \overset{\text{(c)}}{=} 4L + H\big(A_{2,1}^{[(2,1)]}\big|W_1, Z_2\big) + H\big(X_{1,2,2}^{[(2,1)]}\big|W_1\big) \\ & \quad \geq 4L + H\big(A_{2,1}^{[(2,1)]}\big|W_1, Z_2\big) + H\big(A_{1,1}^{[(2,1)]}\big|W_1, Z_2\big) \\ & \qquad + H\big(Z_2|W_1\big) \\ & \quad \geq 4L + H\big(A_{1,1}^{[(2,1)]}, A_{2,1}^{[(2,1)]}|W_1, Z_2\big) + H(Z_2|W_1) \\ & \quad = 4L + H\big(A_{1,1}^{[(2,1)]}, A_{2,1}^{[(2,1)]}, Z_2|W_1\big) \\ & \quad = 4L + H\big(A_{1,1}^{[(1,2)]}, A_{2,1}^{[(1,2)]}, Z_2|W_1\big) \\ & \quad = 5L \end{aligned} \tag{47}$$

where (a) is due to $X_{1,1,1}^{[(1,2)]} \rightarrow W_1$, $X_{2,1,2}^{[(2,1)]} \rightarrow W_1$ and $X_{1,2,2}^{[(2,1)]} \rightarrow W_1$; In (b) we used the chain rule and non-negativity of mutual information, i.e., $H\big(X_{2,1,2}^{[(2,1)]}|W_1\big) = H(Z_2|W_1) + H\big(A_{2,1}^{[(2,1)]}|W_1, Z_2\big) + H\big(A_{1,2}^{[(2,1)]}|W_1, Z_2, A_{2,1}^{[(2,1)]}\big) \geq H\big(A_{2,1}^{[(2,1)]}|W_1, Z_2\big) + H(Z_2|W_1)$; (c) is because both W_1 and W_2 can be decoded from $X_{1,1,1}^{[(1,2)]}$ and Z_2. (47) implies $R_{\text{d}}(M) \geq 5/3 - M$, which completes the converse proof of Theorem 3.

VII. Conclusion

In this paper, we introduced the problem of cache-aided multiuser Private Information Retrieval (MuPIR), which generalizes the single-user cache-aided PIR problem to the case of multiple users. We provided achievability for the MuPIR problem with two messages, two users and arbitrary number of databases utilizing the novel idea of cache-aided Interference Alignment (CIA). The proposed scheme is shown to be optimal when the cache placement is uncoded. For general system parameters, inspired by both single-user PIR and coded caching, we proposed a product design which is order optimal within a factor of 8. Moreover, when the user's demands are constrained to be distinct, the optimal memory-load tradeoff is characterized for a system with two messages, two users and two databases. Due to the strong connection to both PIR and coded caching, our result on the cache-aided MuPIR problem provides useful insights into understanding the role of side information (i.e., cache) in multiuser and multi-message PIR. Besides the proposed achievability and converse results, the cache-aided MuPIR problem still remains open for arbitrary system parameters in terms of the optimal memory-load trade-off. For example, utilizing the idea of CIA, we expect more achievability results to come. Also, based on the well-established converse results of coded caching and PIR, a systematic approach to characterize the converse is needed.

APPENDIX A
COMPUTER-AIDED CONVERSE

In this section, we provide a brief description of the open source toolbox CAI developed by [29], which conducts a computer-aided investigation on the fundamental limits of information systems. By utilizing the linear programming (LP) framework involving the Shannon-type inequalities, the CAI solver is able to read a problem description file which defines the problem-specific random variables and their dependency. It then computes a bound for a given linear combination of information measures, and provides the value of information measures at the optimal solution or a proof as a weighted sum of known information inequalities. This toolbox was shown rather effective in the problems of distributed storage, coded caching and PIR [34]–[36].

We next show how to use the CAI prover to prove the optimality of the proposed CIA based scheme for $K = K_{\mathrm{u}} = 2$ and $N = 2, 3$. Due to limitation of space, we only present the case for $N = 2$. For brevity, we use a different notation from the main paper here.

Sketch of the Computer-Aided Proof

For the $K = K_{\mathrm{u}} = N = 2$ cache-aided MuPIR problem, let f denote an answer of DB 1. By demand privacy, there must exist four different configurations of the DB 2 answer, denoted by g_1, g_2, g_3 and g_4, such that the following decodability condition can be satisfied: $(f, g_1, Z_1) \to W_1$ (meaning that W_1 can be recovered from f, g_1 and Z_1), $(f, g_1, Z_2) \to W_1$, $(f, g_2, Z_1) \to W_1$, $(f, g_2, Z_2) \to W_2$, $(f, g_3, Z_1) \to W_2$, $(f, g_3, Z_2) \to W_1$, $(f, g_4, Z_1) \to W_2$, and $(f, g_4, Z_2) \to W_2$. The following privacy condition

$$H\left(g_i|\mathcal{S}\right) = H\left(g_j|\mathcal{S}\right), \ \forall i, j \in \ [4], \forall \mathcal{S} \subseteq \{Z_1, Z_2, W_1, W_2\} \tag{48}$$

also needs to be satisfied. (48) implies that the four different configurations of DB 2's answer have identical entropy after removing the information $\mathcal{S}$ available to DB 2, i.e., these configurations are indistinguishable from DB 2's viewpoint. We also have the symmetry condition $H(f) = H(g_i)$, $\forall i \in [4]$ and the message and cache memory size constraints $H(W_1) = H(W_2) = 1, H(Z_1) = H(Z_2) = M$. Putting the above constraints as an input file to the CAI prover with LP objective $5R + 6M$, it produces the following points: $(0.000000, 2.000000)$, $(0.250000, 1.500000)$, $(0.666667, 1.000000)$, $(2.000000, 0.000000)$, which coincides with the four achievable corner points by the CIA based scheme. This proves the optimality of the proposed scheme. For $N = 3$, a similar approach can be used.

REFERENCES

[1] X. Zhang, K. Wan, H. Sun, and M. Ji, "Cache-aided multiuser private information retrieval," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2020, pp. 1–6.

[2] X. Zhang, K. Wan, H. Sun, M. Ji, and G. Caire, "Private cache-aided interference alignment for multiuser private information retrieval," in *Proc. 18th Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw. (WiOPT)*, Jun. 2020, pp. 1–8.

[3] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," in *Proc. IEEE 36th Annu. Found. Comput. Sci.*, Oct. 1995, pp. 41–50.

[4] H. Sun and S. A. Jafar, "Blind interference alignment for private information retrieval," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 560–564.

[5] H. Sun and S. A. Jafar, "The capacity of private information retrieval," *IEEE Trans. Inf. Theory*, vol. 63, no. 7, pp. 4075–4088, Jul. 2017.

[6] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.

[7] K. Wan, D. Tuninetti, and P. Piantanida, "An index coding approach to caching with uncoded cache placement," *IEEE Trans. Inf. Theory*, vol. 66, no. 3, pp. 1318–1332, Mar. 2020.

[8] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Characterizing the rate-memory tradeoff in cache networks within a factor of 2," *IEEE Trans. Inf. Theory*, vol. 65, no. 1, pp. 647–663, Jan. 2019.

[9] R. Tandon, "The capacity of cache aided private information retrieval," in *Proc. 55th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Oct. 2017, pp. 1078–1082.

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

[11] Y.-P. Wei, K. Banawan, and S. Ulukus, "Private information retrieval with partially known private side information," in *Proc. 52nd Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2018, pp. 1–6.

[12] S. Kadhe, B. Garcia, A. Heidarzadeh, S. El Rouayheb, and A. Sprintson, "Private information retrieval with side information," *IEEE Trans. Inf. Theory*, vol. 66, no. 4, pp. 2032–2043, Apr. 2020.

[13] A. Heidarzadeh, B. Garcia, S. Kadhe, S. E. Rouayheb, and A. Sprintson, "On the capacity of single-server multi-message private information retrieval with side information," in *Proc. 56th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Oct. 2018, pp. 180–187.

[14] A. Heidarzadeh, F. Kazemi, and A. Sprintson, "Capacity of single-server single-message private information retrieval with private coded side information," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 1662–1666.

[15] A. Heidarzadeh, S. Kadhe, S. El Rouayheb, and A. Sprintson, "Single-server multi-message individually-private information retrieval with side information," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 1042–1046.

[16] F. Kazemi, E. Karimi, A. Heidarzadeh, and A. Sprintson, "Single-server single-message online private information retrieval with side information," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 350–354.

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

[18] S. Li and M. Gastpar, "Converse for multi-server single-message PIR with side information," in *Proc. 54th Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2020, pp. 1–6.

[19] Z. Chen, Z. Wang, and S. A. Jafar, "The capacity of T-private information retrieval with private side information," *IEEE Trans. Inf. Theory*, vol. 66, no. 8, pp. 4761–4773, Aug. 2020.

[20] K. Wan and G. Caire, "On coded caching with private demands," *IEEE Trans. Inf. Theory*, vol. 67, no. 1, pp. 358–372, Jan. 2021.

[21] K. Wan, H. Sun, M. Ji, D. Tuninetti, and G. Caire, "Device-to-device private caching with trusted server," 2019, *arXiv:1909.12748*. [Online]. Available: https://arxiv.org/abs/1909.12748

[22] S. Kamath, "Demand private coded caching," 2019, *arXiv:1909.03324*. [Online]. Available: https://arxiv.org/abs/1909.03324

[23] V. R. Aravind, P. Sarvepalli, and A. Thangaraj, “Subpacketization in coded caching with demand privacy,” 2019, *arXiv:1909.10471*. [Online]. Available: https://arxiv.org/abs/1909.10471

[24] K. Wan, H. Sun, M. Ji, D. Tuninetti, and G. Caire, “On optimal load-memory tradeoff of cache-aided scalar linear function retrieval,” 2020, *arXiv:2001.03577*. [Online]. Available: https://arxiv.org/abs/2001.03577

[25] Q. Yan and D. Tuninetti, “Fundamental limits of caching for demand privacy against colluding users,” 2020, *arXiv:2008.03642*. [Online]. Available: https://arxiv.org/abs/2008.03642

[26] M. A. Maddah-Ali and U. Niesen, “Cache-aided interference channels,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2015, pp. 809–813.

[27] J. Hachem, U. Niesen, and S. N. Diggavi, “Degrees of freedom of cache-aided wireless interference networks,” *IEEE Trans. Inf. Theory*, vol. 64, no. 7, pp. 5359–5380, Jul. 2018.

[28] X. Zhang, N. Woolsey, and M. Ji, “Cache-aided interference management using hypercube combinatorial cache design with reduced subpacketizations and order optimal sum-degrees of freedom,” 2020, *arXiv:2008.08978*. [Online]. Available: https://arxiv.org/abs/2008.08978

[29] C. Tian, J. S. Plank, and B. Hurst. (Oct. 2019). *An Open-Source Toolbox for Computer-Aided Investigation on the Fundamental Limits of Information Systems, Version 0.1*. [Online]. Available: https://github.com/ct2641/CAI/releases/tag/0.1

[30] K. Wan, D. Tuninetti, and P. Piantanida, “On the optimality of uncoded cache placement,” in *Proc. IEEE Inf. Theory Workshop (ITW)*, Sep. 2016, pp. 161–165.

[31] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, “The exact rate-memory tradeoff for caching with uncoded prefetching,” *IEEE Trans. Inf. Theory*, vol. 64, no. 2, pp. 1281–1296, Feb. 2018.

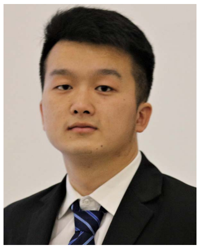
[32] K. Banawan and S. Ulukus, “The capacity of private information retrieval from coded databases,” *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1945–1956, Mar. 2018.

[33] H. Ghasemi and A. Ramamoorthy, “Improved lower bounds for coded caching,” *IEEE Trans. Inf. Theory*, vol. 63, no. 7, pp. 4388–4413, Jul. 2017.

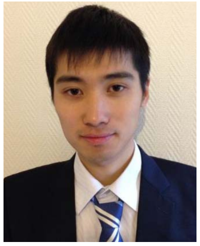
[34] C. Tian, “Characterizing the rate region of the (4,3,3) exact-repair regenerating codes,” *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 967–975, May 2014.

[35] C. Tian, “Symmetry, outer bounds, and code constructions: A computer-aided investigation on the fundamental limits of caching,” *Entropy*, vol. 20, no. 8, p. 603, Aug. 2018.

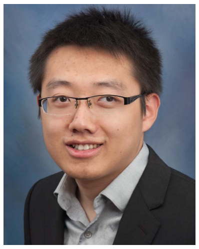
[36] C. Tian, H. Sun, and J. Chen, “A Shannon-theoretic approach to the storage-retrieval tradeoff in PIR systems,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018, pp. 1904–1908.



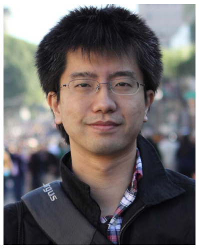
Xiang Zhang (Student Member, IEEE) received the B.E. degree in electronic and information engineering from Xi’an Jiaotong University, Xi’an, China, in 2016, and the degree from the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, in 2017. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, The University of Utah, Salt Lake City, UT, USA. His research interests include information theory, coding theory, wireless communication, information retrieval in caching systems, and distributed computing. He was a recipient of the National Endeavor Scholarship of China in 2015.



Kai Wan (Member, IEEE) received the B.E. degree in optoelectronics from the Huazhong University of Science and Technology, China, in 2012, and the M.Sc. and Ph.D. degrees in communications from Université Paris-Saclay, France, in 2014 and 2018, respectively. He is currently a Post-Doctoral Researcher with the Communications and Information Theory Chair (CommIT), Technische Universität Berlin, Berlin, Germany. His research interests include information theory, coding techniques, and their applications on coded caching, index coding, distributed storage, distributed computing, wireless communications, and privacy and security.



Hua Sun (Member, IEEE) received the B.E. degree in communications engineering from the Beijing University of Posts and Telecommunications, China, in 2011, and the M.S. degree in electrical and computer engineering and the Ph.D. degree in electrical engineering from the University of California at Irvine, Irvine, USA, in 2013 and 2017, respectively. He is currently an Assistant Professor with the Department of Electrical Engineering, University of North Texas, USA. His research interests include information theory and its applications to communications, privacy, security, and storage. He was a recipient of the NSF CAREER Award in 2021. His coauthored papers received the IEEE Jack Keil Wolf ISIT Student Paper Award in 2016 and the IEEE GLOBECOM Best Paper Award in 2016.



Mingyue Ji (Member, IEEE) received the B.E. degree in communication engineering from the Beijing University of Posts and Telecommunications, China, in 2006, the M.Sc. degree in electrical engineering from the Royal Institute of Technology, Sweden, in 2008, and the University of California at Santa Cruz, Santa Cruz, in 2010, and the Ph.D. degree from the Ming Hsieh Department of Electrical Engineering, University of Southern California, in 2015. He subsequently was a Staff II System Design Scientist with Broadcom Corporation (Broadcom Ltd.) from 2015 to 2016. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering and an Adjunct Assistant Professor with the School of Computing, The University of Utah. He is interested in the broad area of information theory, coding theory, concentration of measure and statistics with the applications of caching networks, wireless communications, distributed storage and computing systems, distributed machine learning, and (statistical) signal processing. He received the Best Paper Award in IEEE ICC 2015 Conference, the Best Student Paper Award in IEEE European Wireless 2010 Conference, the USC Annenberg Fellowship from 2010 to 2014, and the IEEE Communications Society Leonard G. Abraham Prize for the Best IEEE JSAC Paper in 2019. He has served as an Associate Editor for IEEE TRANSACTIONS ON COMMUNICATIONS in 2020.

Giuseppe Caire (Fellow, IEEE) was born in Torino in 1965. He received the B.Sc. degree in electrical engineering from the Politecnico di Torino in 1990, the M.Sc. degree in electrical engineering from Princeton University in 1992, and the Ph.D. degree from the Politecnico di Torino in 1994.

From 1994 to 1995, he was a Post-Doctoral Research Fellow with the European Space Agency, ESTEC, Noordwijk, The Netherlands. He was an Assistant Professor in telecommunications with the Politecnico di Torino, an Associate Professor with the University of Parma, Italy, a Professor with the Department of Mobile Communications, Eurecom Institute, Sophia-Antipolis, France, and a Professor of electrical engineering with the Viterbi School of Engineering, University of Southern California at Los Angeles, Los Angeles. He is currently an Alexander von Humboldt Professor with the Faculty of Electrical Engineering and Computer Science, Technical University of Berlin, Germany. His main research interests include the field of communications theory, information theory, channel and source coding, with particular focus on wireless communications. He was the President of the IEEE Information Theory Society in 2011. He has served on the Board of Governors for the IEEE Information Theory Society from 2004 to 2007 and as an Officer from 2008 to 2013. He received the Jack Neubauer Best System Paper Award from the IEEE Vehicular Technology Society in 2003, the IEEE Communications Society and Information Theory Society Joint Paper Award in 2004 and in 2011, the Okawa Research Award in 2006, the Alexander von Humboldt Professorship in 2014, the Vodafone Innovation Prize in 2015, the ERC Advanced Grant in 2018, the Leonard G. Abraham Prize for Best IEEE JSAC Paper in 2019, the IEEE Communications Society Edwin Howard Armstrong Achievement Award in 2020. He was a recipient of the 2021 Leibniz Prize of the German National Science Foundation (DFG).