Coded Caching With Private Demands and Caches

Ali Gholami, *Student Member, IEEE*, Kai Wan, *Member, IEEE*, Hua Sun, *Member, IEEE*, Mingyue Ji, *Member, IEEE*, and Giuseppe Caire, *Fellow, IEEE*

***Abstract*— This paper investigates the privacy problem in coded caching. Recently, it was shown that the seminal MAN coded caching scheme leaks the demand information of each user to the other users in the system. Many works have considered coded caching with demand privacy, while every non-trivial existing coded caching scheme with private demands was built on the fact that the cache information of each user is private to the others. However, most of these schemes leak the users' cache information. As a consequence, in most realistic settings (e.g., video streaming) where the system is used over time with multiple sequential transmission rounds, these schemes leak demand privacy beyond the first round. This observation motivates our new formulation of coded caching with simultaneously private demands and caches in this paper. For this new model, we first show that an existing coded caching scheme with private demands, referred to as the *virtual users* scheme, can also preserve the privacy of the users' caches. However, this scheme suffers from its extremely high subpacketization. The main contribution of this paper is a new construction that generates private coded caching schemes by leveraging two-server private information retrieval (PIR) schemes. We show that if in the PIR scheme the demand is uniform over all files and the queries are independent, the resulting caching scheme is private on both the demands and the caches; otherwise, the resulting scheme is private only on the demands. This first result constructs coded caching schemes from a particular class of PIR schemes, which is a new "structural" result in its own merit. We then construct new two-server PIR schemes with uniform demand and independent queries, such that the resulting caching scheme has a subpacketization level that is significantly reduced compared to the virtual users scheme. Interestingly we propose a new construction of two-server PIR schemes with uniform demand and independent queries by exploiting coded caching schemes. By applying the seminal Maddah-Ali and Niesen coded caching scheme in our construction, the resulting two-server PIR scheme is proved to be order-optimal under the constraint of uniform demand and independent queries. This is a second new "structural" result that somehow closes the loop in the relationship between coded caching and PIR. As a by-product of our new construction, we obtain a new demand private that improves the load of the state-of-the-art demand private caching schemes known so far. Finally, to explore a broader tradeoff between cache privacy and transmission load, we relax the cache privacy constraint and introduce the definition of cache information leakage. Then, again as a by-product of our new construction, we propose new schemes with perfect demand privacy and imperfect cache privacy that achieve an order-gain in load with respect to the scheme with perfect privacy on both demands and caches. This also establishes a first non-trivial achievability result in the tradeoff between load and cache privacy, for demand-private caching schemes.**

***Index Terms*— Coded caching, private demands and caches, private information retrieval.**

I. Introduction

CODED caching was first introduced in [2]. In a caching system, the goal is to leverage the local memory available at the end-users to reduce the load in the network by exploiting the content already availbale in the cache instead of downloading it from the server. Before the advent of [2], this leverage was limited to the *local caching gain*, which depends on the local cache size. The coded caching scheme proposed by Maddah-Ali and Niesen, referred to as the MAN scheme in [2], showed that the cache memory available to each user can be used in an aggregated manner even if there is no cooperation between the users. This gain is called the *global caching gain*. Thus, in addition to benefiting from a local cache size, the system can benefit from the aggregate cache size which scales with the number of users and provides a greater further reduction in network load.

In the MAN coded caching setting [2], a server which has a library of N files and is connected to K users via a shared medium. Each user has a cache of size M files. A coded caching scheme consists of two phases: *placement* and *delivery*. In the placement phase, each user fills its cache without knowledge of the users' later demands. If each user stores some bits of files directly into the cache, the placement phase is called *uncoded*. In the delivery phase, each user demands one file. According to the users' demands and caches, the server broadcasts multicast messages to the users such that

Manuscript received 19 December 2022; revised 26 September 2023; accepted 16 November 2023. Date of publication 28 November 2023; date of current version 22 January 2024. The work of Ali Gholami and Giuseppe Caire was supported by the European Research Council through the ERC Advanced, CARENET, under Grant 789190. The work of Kai Wan was supported in part by the National Natural Science Foundation of China under Grant NSFC-12141107 and in part by the CCF-Hikvision Open Fund under Grant 20210008. The work of Hua Sun was supported in part by NSF under Grant CCF-2007108, Grant CCF-2045656, and Grant CCF-2312228. The work of Mingyue Ji was supported in part by NSF under Award 1817154 and Award 1824558 and in part by NSF CAREER under Grant 2145835. An earlier version of this paper was presented in part at the 2022 IEEE International Symposium on Information Theory [DOI: 10.1109/ISIT50566.2022.9834846]. *(Corresponding author: Kai Wan.)*

Ali Gholami and Giuseppe Caire are with the Department of Electrical Engineering and Computer Science, Technical University of Berlin, 10587 Berlin, Germany (e-mail: a.gholami@tu-berlin.de; caire@tu-berlin.de).

Kai Wan was with the Department of Electrical Engineering and Computer Science, Technical University of Berlin, 10587 Berlin, Germany. He is now with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: kai_wan@hust.edu.cn).

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

Mingyue Ji is with the Department of Electrical and Computer Engineering, University of Utah, Salt Lake City, UT 84112 USA (e-mail: mingyue.ji@utah.edu).

Communicated by C. Hollanti, Associate Editor for Coding and Decoding.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIT.2023.3336792.

Digital Object Identifier 10.1109/TIT.2023.3336792

each user can retrieve its demanded file. The transmission load is defined as the number of bits broadcasted in the delivery phase normalized by the file size. The objective is to minimize the worst case load among all possible demands. The MAN coded caching scheme is based on a combinatorial design in the placement which splits each file into multiple subfiles and assigns each subfile to a subset of users, such that each multicast message is useful to $1 + KM/N$ users. The achieved load by the MAN coded caching scheme is $\frac{K(1-M/N)}{1+KM/N}$, where $1 - M/N$ represents the *local* caching gain and $\frac{1}{1+KM/N}$ represents the *global* caching gain. When $N \geq K$, the MAN scheme was proved to be order-optimal within a factor of 2 [3] and optimal under the constraint of uncoded cache placement [4]. When $N < K$, an improved coded caching scheme was proposed by Yu, Maddah-Ali, and Avestimhr (YMA) in [3], which relies on the fact that some MAN multicast messages can be reconstructed by the other ones and are thus redundant. The YMA scheme was then proved to be order-optimal within a factor of 2 [3], and optimal under the constraint of uncoded cache placement [3], for arbitrary system parameters. Following the seminal work of MAN, coded caching has been studied in various extensions, including decentralized setting [5], online coded caching [6], Device-to-Device (D2D) networks [7], random and non-uniform demands [8], [9], hierarchical coded caching [10], etc.

The MAN centralized scheme has a subpacketization level at most exponential in the number of users K, which is one of its practical limitations. Also for their decentralized algorithm in [5], the multiplicative caching gain appears in the asymptotic regime of file size scaling to infinity. The authors in [11] addressed this issue and showed that this multiplicative gain is non-existent in the finite file size regime under random placement and clique cover delivery schemes. To reduce subpacketization, the authors in [12] introduce a decentralized scheme that achieves a low worst-case load in the finite file size regime and maintains an optimal memory-load tradeoff when file size scales to infinity. A combinatorial structure, referred to as placement delivery array (PDA), was proposed in [13] to design coded caching schemes with uncoded cache placement and clique-covering delivery, where the MAN scheme can also be seen as a coded caching scheme under PDA construction. Following [13], various PDA constructions have been proposed in [14], [15], [16], [17], and [18]. Other combinatorial structures, such as hypergraphs [19], Ruzsa-Szeméredi graphs [20], the strong edge coloring of bipartite graphs [21], linear block codes [22], have also been used to construct coded caching schemes with reduced subpacketization compared to the MAN scheme. Under PDA construction, the subpacketization of the MAN scheme is minimal to achieve the load $\frac{K(1-M/N)}{1+KM/N}$ [23].

A. Demand Private Coded Caching Schemes

Despite the optimality guarantee, another problem of the MAN scheme is the leakage of users' demand information. In order to decode the MAN multicast messages, each user should be aware of the other users' demands, which violates the demand privacy. An information-theoretic formulation of coded caching with private demands has been proposed in [24], where each user has a cache that is private to the other users, and the privacy constraint requires that each user cannot obtain any information about other users' demands from the broadcast messages in the delivery phase. Based on the virtual user strategy in [25], an information-theoretic private scheme has been proposed in [24]. By introducing $KN - K$ virtual users and letting each file be demanded by K effective users (i.e., real or virtual users), the problem can be solved by using the NK-user MAN or YMA scheme. The resulting scheme can perfectly preserve the privacy of each user's demand against the other users, because each user cannot distinguish the real users from the effective users. The achieved load of this virtual user-based scheme has been proved to be order-optimal within a constant, except for the case $N \geq K$ and $M < N/K$. However, its subpacketization is at most exponential in NK, which is far from being practical. To reduce the subpacketization, the authors in [24] proposed another private caching scheme based on Minimum Distance Separable (MDS) codes for the case $M \geq N/2$, which achieves an order-optimal load with subpacketization at most exponential in K. A different (but equivalent) scheme based on virtual users has been proposed in [26], where for each real user we introduce $N-1$ virtual users, such that the union set of files requested by these N effective users is the entire library.

Following the coded caching problem with private demands, some improved schemes have been proposed in order to reduce the load or subpacketization. In [27], the authors proposed a demand-private scheme for the special case of a caching system with $N = 2, K = 2$ and $M = 1$, while the subpacketization level equal to 3 was proved to be the minimum. A strategy introducing the use of *private keys* has been proposed by Yan and Tuninetti in [28], whose main idea is to transform file retrieval to scalar linear function retrieval (i.e., each user requests a scalar linear function of files [29]). Each user's cache is divided into two parts. In the first part, each user caches the same subfiles as in the MAN scheme. The second part serves as each user's private key, composed of some linear combinations of the subfiles which are not cached in the first part. In the delivery phase, each user pretends to request a scalar linear function of the files, from which and the private key the user can recover its demanded file. Then, by using the cache-aided scalar linear function retrieval scheme in [29], the resulting private caching scheme requires a subpacketization which is the same as the MAN scheme. The resulting scheme was proved to be order-optimal within a factor of 6.3707 when the metadata (i.e., the composition) of the broadcast messages is given.[1] Other works on demand private caching include [30], which proves that the optimal loads with and without demand privacy are within a multiplicative factor, and also characterizes the exact memory-load tradeoff for the case $N = K = 2$. Furthermore, in [31], the authors provided the exact memory-load tradeoff for demand private

[1]In the privacy constraint of [28], the mutual information is under the condition of the realization of the library; this is equivalent to the case where the metadata of the broadcast messages is provided in the header of the delivered packet, which is common in practice for the ease of decoding.

coded caching when $N \geq K = 2$. Finally in [32], demand private coded caching was studied with a focus on reducing the subpacketization level. For the cases $N = K = 2$, the authors proposed a scheme with the lowest possible subpacketization.

B. Brief Review of Private Information Retrieval (PIR)

Demand privacy was originally considered in the PIR problem [33], where a user is connected to S servers through S individual private links, respectively. The library contains N equal-length messages, and the user wants to retrieve one message from the servers without letting the servers know any information about the demand. For this purpose, the user sends a query to each server, and the server replies with an answer containing some coded packets to the user. The communication cost, defined as the amount of information exchanged between the user and the servers, is equal to the sum of the total upload cost (i.e., the sum of the individual upload costs defined as the length of the query from the user to the servers) and the total download cost (i.e., the sum of the individual download costs defined as the length of the answer from the servers to the user normalized by the message size).

For the single-server PIR problem, the only solution that preserves the information-theoretic demand privacy is to download the entire library. Numerous works have considered minimizing of the communication cost for the system with multiple servers. A two-server PIR scheme with communication cost of $O(N^{1/3})$ was proposed in [33] based on covering codes [34], which was then extended to the S-server system with communication cost $O\left(N^{1/(2S-1)}\right)$ [35]. Also in [36], the author introduced a time-efficient two-server PIR with the same communication complexity of $O(N^{1/3})$ as in [33]. In [37], the authors considered the problem of t-private PIR, where the goal is to keep the identity of the demanded file private, even with the collusion of up to t servers. Some other important works on PIR include [38], [39], [40], [41], where the authors study the bounds on communication cost. The work in [42] is a polynomial-based approach and achieves the $O(N^{1/3})$ communication cost and the work in [43], introduces the currently best known communication cost of $N^{o(1)}$ for two-server PIR schemes and is based on the polynomial approach of [42] and matching vector codes (MVC) [44], [45].

Due to the difficulty of characterizing the optimal communication cost, another direction on the PIR problem is to characterize the optimal total download cost. In [46] Sun and Jafar characterized the optimal total download cost, $1 + 1/S + 1/S^2 + \cdots + 1/S^{N-1}$, by proposing an interference alignment-type achievable scheme and a matching converse. In [47], the authors introduce an asymmetric PIR scheme that achieves the optimal total download cost. Furthermore, under the constraint of achieving the optimal total download cost, this scheme has the minimum total upload cost $S(N-1)\log_2 S$ and the minimum subpacketization on the message $S-1$. In addition, some extended PIR models were considered with the objective to minimize the total download cost, including multi-message PIR (where the user wants to privately retrieve M messages from the servers) [48], symmetric PIR (where there is an additional security constraint that the user cannot receive any information about the undesired messages) [49], PIR with side information (where the user has some prior side information in the form of a subset of messages not including the desired one) [50], PIR from MDS-coded data in distributed storage systems [51], cache-aided PIR (where the user has a cache storage that can be used to store any function of the messages) [52], multi-message PIR with private side information (where the identity of the desired messages and the side information should be kept private from the servers) [53], PIR with colluding databases (where a number of databases may share the received queries among each other) [54], and PIR with coded databases [55].

C. Contributions

In the coded caching problem with private demands [24], an important condition for designing non-trivial private caching schemes is that each user's cache information is private to the others; otherwise, to preserve the demand privacy we need to let each user decode the entire library. However, in most existing private caching schemes (except the virtual users scheme in [26]), the users' caches are leaked after the transmission in the delivery phase; thus, after one transmission round in which each user has decoded one file, these schemes cannot be used to preserve the demand privacy when each user wants to retrieve another file in a new transmission round. This motivates the formulation of the coded caching problem with private demands and caches in this paper: in addition to the privacy constraint on the users' demands, we also want to preserve the privacy of the users' caches. Besides formulating of this new problem, our contributions are as follows.

- We first show that the virtual users scheme in [26] is private in terms of demands and caches. The achieved load of this scheme is order-optimal within a constant factor except for the case where $N > K$ and $M < N/K$. However, the subpacketization of this scheme is $2^{\mathcal{H}(M/N)NK}$ and is at most exponential in NK, where $\mathcal{H}(\cdot)$ represents the binary entropy function.
- In order to reduce the subpacketization of the virtual users scheme, we propose a new construction structure on private coded caching schemes by leveraging two-server PIR schemes. In particular, we show that the schemes resulting from our construction are demand private. By applying the PIR scheme in [47], we can construct a demand-private coded caching scheme with an improved memory-load tradeoff than that of [28]. We then show that if the underlying PIR scheme has the uniform demand and independent query (UDIQ) property (see Definition 1 in Section II-C), the resulting caching scheme is both demand- and cache-private. These first results introduce a new "structural" result that constructs the coded caching schemes from a particular class of PIR schemes.
- As a consequence of the above result, we then shift our focus to the construction of two-server PIR schemes with the UDIQ property. Interestingly, we find a new construction structure for two-server PIR schemes under the UDIQ condition by leveraging coded caching schemes.

By applying the Maddah-Ali and Niesen scheme to our construction, the achieved load by the resulting two-server PIR scheme is proved to be order-optimal under the UDIQ constraint. This is a second new "structural" result that somehow closes the loop in the relationship between coded caching and PIR.

- In order to explore a broader tradeoff between subpacketization order, transmission load, and cache privacy, we relax the UDIQ constraint, and as a result, obtain demand private coded caching schemes with a controlled amount of leakage on the cache information, which opens the path in this new exploration. In particular, using the PIR scheme in [43], we obtain a demand private coded caching scheme with better cache information leakage than [28]. Recall that using the PIR scheme in [47], we obtain a demand private coded caching scheme achieving lower load than [28] with the same subpacketization. These results clearly demonstrate the flexibility of our construction.

D. Paper Organization

The rest of this paper is organized as follows. The system model is presented in Section II. Section III presents our main results on coded caching with private demands and caches. Section IV presents the results for the extended model where some leakage on the caches is allowed. We conclude the paper in Section V.

E. Notation Convention

Calligraphic symbols denote sets, bold symbols denote vectors, and sans-serif symbols denote system parameters. We denote the set $\{a, a+1, \ldots, b\}$ by $[a:b]$ and $[b]$ refers to $[1:b]$. We use $|\cdot|$ to denote the cardinality of a set or the length of a vector. Also, $B_{\mathcal{A}}$ denotes the set $\{B_i, \forall i \in \mathcal{A}\}$. The base of logarithm in this paper is 2.

II. System Model

A. Problem Formulation of Coded Caching With Private Demands and Caches

The considered coded caching system consists of a server with access to a library of N independent files denoted by $W_1, W_2, \ldots, W_N$. This server is connected to K cache-aided users via a shared link. The entropy of the each user's cache content is limited by MF. We assume that each file has F bits. The system operates in two phases.

Placement Phase: Each user fills its cache without knowledge of later demands. The cached content of user $k \in [K]$ is

$$Z_k = \phi_k(W_1, \ldots, W_N, \mathscr{M}_k), \tag{1}$$

where $\mathscr{M}_k$ represents the metadata of the bits in Z_k. $\mathscr{M}_k$ is a random variable over $\mathscr{C}_k$, representing all types of cache placements of user k. The realization of $\mathscr{M}_k$ is known only by the server and user k. The memory size constraint states that the cache size should be MF, i.e.,

$$H(Z_k) \leq MF, \ \forall k \in [K].^{2} \tag{2}$$

Following the assumption made in [24], we assume that F is sufficiently large such that the size of $\mathscr{M}_k$ is negligible with respect to the file size, and $\mathscr{M}_k$ is also provided in Z_k.

Delivery Phase: During the delivery phase, user $k \in [K]$ requests one file W_{d_k}, where d_k is uniformly i.i.d. over $[N]$. The demanded vector is denoted by $\mathbf{d} = (d_1, d_2, \ldots, d_K)$. Given the demand vector $\mathbf{d}$, the server broadcasts to all users

$$X_{\mathbf{d}} = \psi(\mathbf{d}, W_1, \ldots, W_N, \mathscr{M}_1, \ldots, \mathscr{M}_K). \tag{3}$$

Note that we have

$$H(W_{[N]}, \mathscr{M}_{[K]}, \mathbf{d}) = NF + H(\mathscr{M}_{[K]}) + \sum_{k\in[K]} H(d_k). \tag{4}$$

We also assume that the metadata of the broadcast message is contained within the message and is negligible compared to the file size.

Decoding: User $k \in [K]$ decodes its desired file W_{d_k} from $\big(d_k, Z_k, X_{\mathbf{d}}\big)$, i.e.,

$$H\big(W_{d_k} | d_k, Z_k, X_{\mathbf{d}}\big) = 0. \tag{5}$$

Privacy: We want to preserve the privacy of each user's demand against other users, i.e.,

$$I(\mathbf{d}; X_{\mathbf{d}} | d_k, Z_k) = 0, \ \forall k \in [K]. \tag{6}$$

In addition to (6), we want to preserve the privacy of the metadata of each user's cache content against other users, i.e.,

$$I\big((\mathscr{M}_1, \ldots, \mathscr{M}_K); X_{\mathbf{d}} | d_k, Z_k\big) = 0, \ \forall k \in [K]. \tag{7}$$

Objective: The load R is achievable if there exist cache placement functions $\{\phi_k(\cdot) : k \in [K]\}$, encoding function $\psi(\cdot)$, and decoding functions $\{\theta_k(\cdot) : k \in [K]\}$ such that

$$W_{d_k} = \theta_k(d_k, Z_k, X_{\mathbf{d}}), \forall k \in [K], \tag{8}$$

$$\text{where } H(X_{\mathbf{d}})/F \leq R. \tag{9}$$

Our objective is to find the minimum achievable load $R^\star$ for given system parameters M, N, K,[3] i.e.,

$$R^\star = \min_{\phi_k, \psi, \theta_k : k\in[K]} R. \tag{10}$$

[2]The number of bits per information file symbol to represent cache $Z_k = z_k$ is $(1/F) * \lceil \log_2(1/\Pr(Z_k = z_k)) \rceil$. Thus the average number of bits to represent the cache is $(1/F) \sum_{z_k} \lceil \log_2(1/\Pr(Z_k = z_k)) \rceil * \Pr(Z_k = z_k) \simeq H(Z_k)/F$, where the error (rounding integer) is $O(1/F)$. Hence, for large F we can neglect such rounding, and impose a constraint on the cache entropy, $H(Z_k) \leq MF$. This does not mean that every realization of Z_k can be represented by MF bits, however, on average over the ensemble of the realizations; this holds for each user k.

[3]Note that the broadcast messages for different demands have the same size, by the constraint of private demands.

B. Review of the Existing Schemes for Coded Caching With Private Demands

Note that if we remove the private constraint on the caches in (7), the considered problem reduces to the coded caching problem with private demands in [24]. In the following, we briefly review two efficient existing private demand coded caching schemes, which are based on the virtual-user strategy and the privacy key strategy, respectively. In the virtual-user strategy proposed in [30], a (N, K, M)-private scheme is constructed using a (N, NK, M)-non-private scheme. In the placement phase, user k's cache encoding function, cache_k, in the private scheme is given by $\text{cache}_k = \text{cache}^{np}_{(k-1)N+S_k}$ for S_k chosen uniformly random from $[N]$ where cache^{np}_i is the cache encoding function of the non-private scheme for user i. The memory-load tradeoff in this scheme is given by the piecewise linear function connecting the memory-load points $\left(\frac{t}{K}, \frac{\binom{NK}{t+1}-\binom{NK-N}{t+1}}{\binom{NK}{t}}\right)$, $\forall t \in [0 : NK]$. Regarding cache privacy, since the choice of S_k is unknown to users other than k, the cache contents of the users are private. In the delivery phase, the assignment of files demanded by the virtual users is such that all N file indices are requested by users $[(k-1)N+1 : kN]$ for each $k \in [K]$. Regarding privacy, the mapping of caches to demands are revealed during server transmission, but the cache-demand pair for real user $k \in [K]$ is not distinguishable among users $[(k-1)N+1 : kN]$, and thus, both caches and demands are private.

The privacy key scheme proposed in [28] does not provide full privacy for users' caches, but keeps the demands private. The placement phase is similar to the MAN scheme and the subfiles cached for each user in the MAN scheme are also cached here, but in addition, a linear function of subfiles for each uncached subfile index is stored in the cache for each user. The coefficients of this linear combination are chosen randomly by each user and kept private from others. In the delivery phase, based on these coefficients, the user requests a linear function of subfiles so that the retrieval of the demanded subfiles are possible. If the coefficient vector $\mathbf{p}_k$ is used in the placement phase for user k, then in the delivery phase the requested coefficient vector would be $\mathbf{p}_k + \mathbf{d}_k$, where $\mathbf{d}_k$ has a 1 in position d_k (the demanded file index of user k) and 0 elsewhere. The memory-load tradeoff for this scheme is given by the piecewise linear function connecting the memory-load points $\left(1 + \frac{t(N-1)}{K}, \frac{\binom{K}{t+1}-\binom{K-\min\{N-1,K\}}{t+1}}{\binom{K}{t}}\right)$, $\forall t \in [0 : K]$. Regarding privacy, since $\mathbf{p}_k$ is a uniformly chosen random vector on $\{0,1\}^N$, $\mathbf{p}_k + \mathbf{d}_k$ would also be uniformly distributed on $\{0,1\}^N$ regardless the choice of $\mathbf{d}_k$ and as a result, the demands are kept private.

C. Review of Private Information Retrieval

Since our main result is built on a newly discovered connection between private caching schemes and PIR, in this section we review the PIR problem setting.

Assume that there are S servers each containing of a library of N files with B bits, denoted by $W_1, W_2, \ldots, W_N$. A user is connected to these S servers through S individual and private links (i.e., the servers do not collude), and wishes to retrieve one file from the library while keeping the privacy of the demand against the servers. Assuming that the desired file is W_d, for each $s \in [S]$, the user sends the query $Q_s^{[d]} \in \mathcal{Q}_s$ to server s. Based on the received query, server s sends back to the user the answer $A_s^{[d]}$ as a function of the query and the files $W_1, W_2, \ldots, W_N$, i.e.,

$$A_s^{[d]} = \gamma_s(Q_s^{[d]}, W_1, W_2, \ldots, W_N), \tag{11}$$

where γ_s represents the encoding function of server s. Based on the set of answers and queries, there should exist a decoding function by which the user can recover the desired file, i.e.

$$H\left(W_d | A_1^{[d]}, \ldots, A_S^{[d]}, Q_1^{[d]}, \ldots, Q_S^{[d]}\right) = 0. \tag{12}$$

Additionally, the privacy constraint states that the query sent to each server, should not reveal any information about the desired file index; i.e., for each $s \in [S]$,

$$I(d; Q_s^{[d]} | W_1, \ldots, W_N) = 0. \tag{13}$$

From (13) we can conclude that $H(A_s^{[1]}) = \cdots = H(A_s^{[N]}) := H(A_s)$ holds for every $s \in [S]$. The total download cost of the PIR scheme is defined as the total size of information received from the servers over the message size, denoted by $R_D = \frac{\sum_{s\in[S]} H(A_s)}{B}$. The objective of the PIR problem is to characterize the minimum total download cost R_D.[4]

The optimal total download cost has been solved for general system parameters in [46]. This result is recalled here in the following:

Theorem 1 (Capacity of PIR [46]): For the PIR problem with N messages and S databases, the optimal total download cost is

$$1 + 1/S + 1/S^2 + \cdots + 1/S^{N-1}. \tag{14}$$

Interestingly, the optimal total download cost can be achieved not only in the asymptotic regime of arbitrarily large file size. In fact, it is sufficient that the file size is equal to any integer multiple of S^N bits. In [47] the authors proposed a PIR scheme that achieves the optimal (least) file size and total upload cost among the class of decomposable codes that achieve the optimal total download cost. According to [47], the term "decomposable" restricts each coded symbol to be a summation of the component functions on the individual messages. For the exact definition, see [47, Definitions 2 and 3]. Their result is given in the following theorem.

Theorem 2 [47]: Among all download cost optimal uniformly decomposable PIR codes, the PIR code proposed in [47] has the smallest message size, which is $S-1$. Among all download cost optimal decomposable PIR codes, this scheme has the lowest total upload cost, which is $S(N-1)\log_2 N$.

Finally, we introduce the *uniform demand and independent queries* (UDIQ) condition on PIR schemes which will be

[4]Note that in most information-theoretic works on PIR, the objective is to maximize the download rate, which is defined as $\frac{B}{\sum_{s\in[S]} H(A_s)}$. In other words, the download rate is the reciprocal of the total download cost considered in this paper.

needed in our construction of coded caching schemes with private demands and caches.

Definition 1 (UDIQ Condition): For a two-server PIR scheme, if the demand is uniformly distributed over $[N]$ and

$$I(Q_1^{[n]}; Q_2^{[n]} | W_1, \ldots, W_N) = 0, \ \forall n \in [N], \tag{15}$$

then the PIR scheme satisfies the UDIQ condition.

III. Main Results

In this section, we will present our main results on the coded caching problem with private demands and caches. We first show that the virtual users scheme reviewed in Section II-B can also preserve the privacy of the users' caches.

Theorem 3: For the coded caching problem with private demands and caches, $R^\star$ is upper bounded by the lower convex envelop of the following memory-load tradeoff points,

$$\left(\frac{t}{K}, \frac{\binom{NK}{t+1} - \binom{NK-N}{t+1}}{\binom{NK}{t}} \right), \ \forall t \in [0 : NK]. \tag{16}$$

Proof: The demand privacy constraint in (6) has already been proved to hold in [26]. To complete the proof, we need to show that the cache privacy constraint also holds. The virtual users scheme for parameters (N, K, M) is built on the non-private MAN scheme for parameters (N, NK, M) when the demands of virtual users are carefully selected. In this scheme, user k behaves as user $\big((k-1)N + S_k\big)$ in the (N, NK, M) non-private scheme, where $S_k \sim \text{Unif}\{[N]\}$. So the metadata of the cache content of user k is determined by S_k, or equivalently $\mathscr{M}_k = S_k$. In this scheme, the users $(k-1)N+1, (k-1)N+2, \ldots, kN$ cover all N possible demands. Following the demand construction of the (N, NK, M) non-private scheme in [26], define C_k as follows,

$$C_k := (S_k - d_k) \bmod N, k \in [K] \tag{17}$$

Then, let q_k be the right cyclic shift of the vector $(1, \ldots, N)$ by C_k positions. Thus the demand vector of the (N, NK, M)-non-private scheme is $\mathbf{d}^{np} = (\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_K)$. So we can see that the demand vector in the non-private scheme is a function of $\mathbf{C} := (C_1, C_2, \ldots, C_K)$. The transmission of the server for one part should contain the vector $\mathbf{C}$ in order for the users to be able to decode their messages [26]. The other part of the transmission consists of a non-private (N, NK, M) coded caching scheme based on the scheme in [3] which is a function of the library and $\mathbf{d}^{np}$, and since $\mathbf{d}^{np}$ is a function of $\mathbf{C}$, we denote this part of transmission as $X^{np}(W_{[K]}, \mathbf{C})$. So in the end we can write $X_{\mathbf{d}} = (\mathbf{C}, X^{np}(W_{[K]}, \mathbf{C}))$. Now we can write the cache privacy constraint in (7) as follows,

$$\begin{aligned} &I\big((\mathscr{M}_1, \ldots, \mathscr{M}_K); X_{\mathbf{d}} | d_k, Z_k\big) \\ &= I\big(S_1, \ldots, S_K; \mathbf{C}, X^{np}(W_1, \ldots, W_K, \mathbf{C}) | d_k, Z_k\big) && \text{(18a)} \\ &= I(S_1, \ldots, S_K; \mathbf{C} | d_k, Z_k) && \text{(18b)} \\ &+ I(S_1, \ldots, S_K; X^{np}(W_1, \ldots, W_K, \mathbf{C}) | d_k, Z_k, \mathbf{C}) && \text{(18c)} \end{aligned}$$

Based on (17) and the fact that the demands are uniformly distributed, the distribution of $\mathbf{C}$ does not change depending on knowing or not knowing the value of the vector $(S_1, \ldots, S_K)$. Thus the term in (18b) is zero and since $\mathbf{C}$ is already in the condition in (18c), $X^{np}(W_1, \ldots, W_K, \mathbf{C})$ would not have any connection to $(S_1, \ldots, S_K)$ and this term is also zero. Therefore, both the privacy constraints (6) and (7) are satisfied and decodability in (5) is already proved to hold in [26]. This completes the proof. □

Note that it was proved in [24] that the multiplicative gap between the achieved load by the virtual users scheme and the converse bound of the non-private coded caching problem is at most 8, except in the case of $N < K$ and $M < N/K$. This order-optimality result also holds for the considered coded caching problem with private demands and caches.

A. New Construction on Coded Caching With Private Demands

The subpacketization of the virtual users scheme in Theorem 3 is $2^{\mathcal{H}(M/N)NK}$ and is at most exponential in NK, while the subpacketization of the MAN scheme is $2^{\mathcal{H}(M/N)K}$ and is at most exponential in K. Next, we aim to reduce the subpacketization of the virtual users scheme while keeping demand and cache information private simultaneously. The key contribution of this paper is to propose a new construction strategy for private coded caching, which establishes a new relationship between two-server PIR schemes and private coded caching. We first consider demand privacy, and propose a structure in the following theorem to construct demand private coded caching schemes from PIR schemes. The proof is given in Appendix A.

Theorem 4 (From PIR to Coded Caching): Given any two-server PIR scheme with N files and download cost pair (R_{D_1}, R_{D_2}) where R_{D_i} corresponds to server i, there exists an (N, K) coded caching scheme (N files and K users) with private demands whose achieved memory-load tradeoff is the lower convex envelope of $(0, N)$,

$$\begin{aligned} &\Bigg(\frac{Nt}{K} + \big(1 - \frac{t}{K}\big) \left(\mu_1 R_{D_1} + \mu_2 R_{D_2}\right), \\ &\left(\mu_1 R_{D_2} + \mu_2 R_{D_1}\right) \frac{K-t}{t+1} \Bigg), \forall t \in [0 : K-1], \end{aligned} \tag{19}$$

and $(N, 0)$, where $\mu_1, \mu_2 \in [0, 1], \mu_1 + \mu_2 = 1$. Assuming that the needed subpacketization of the given PIR scheme is F', then the needed subpacketization for each point in (19) with $t \in [0 : K-1]$ is $\binom{K}{t} F'$.

Since based on Theorem 4 we are allowed to use any two-server PIR scheme, we can choose the one in [47] which has the optimal total download cost $R_D^\star := 1 + 1/2 + (1/2)^2 + \cdots + (1/2)^{N-1}$ and subpacketization level of $F' = 1$. Therefore, using the scheme in [47] in Theorem 4 (for $\mu_1 = \mu_2 = 1/2$), we will have the following result.

Corollary 1: For the (N, K) coded caching problem with private demands in [24], there exists a scheme whose achieved memory-load tradeoff is the lower convex envelope of $(0, N)$,

$$\begin{aligned} &(M, R) = \\ &\left(\frac{Nt}{K} + \big(1 - \frac{t}{K}\big) \frac{R_D^\star}{2}, \frac{R_D^\star}{2} \frac{K-t}{t+1} \right), \forall t \in [0 : K-1], \end{aligned} \tag{20}$$

and $(N, 0)$. The needed subpacketization for each point in (20) with $t \in [0 : K-1]$ is $\binom{K}{t}$.

Remark 1 (Comparison to the Demand Private Caching Scheme in [28]):

In Theorem 4 for the time-sharing parameters $\mu_1 = \mu_2 = 1/2$ (or the case where $R_{D_1} = R_{D_2}$), the points in (19) become

$$(M, R) = \left(\frac{Nt}{K} + \left(1 - \frac{t}{K}\right)\frac{R_D}{2}, \frac{R_D}{2}\frac{K-t}{t+1} \right), \ \forall t \in [0 : K], \tag{21}$$

where $R_D = R_{D_1} + R_{D_2}$ represents the total download cost. The memory-load tradeoff for the demand private scheme of [28] for the case $K \leq N + t$, follows $(M, R) = \left(\frac{Nt}{K} + \left(1 - \frac{t}{K}\right), \frac{K-t}{t+1} \right)$, which is order-optimal within a constant gap. So when $K \leq N + t$, the achieved memory-load tradeoff in (21) is strictly better than [28] if the chosen PIR scheme has the total download cost $R_D = R_{D_1} + R_{D_2} < 2$. In this case, the resulting scheme is also order-optimal within a constant gap. When $K \leq N + t$, the demand private coded caching scheme in [28] is a special case of our construction in Theorem 4; by applying the two-server PIR scheme in [56] into Theorem 4, the resulting coded caching scheme with private demands becomes the privacy key scheme in [28]. Since the total download cost of the two-server PIR scheme in [47] is strictly less than 2, when $K \leq N + t$, the proposed caching scheme in Corollary 1 has a strictly better performance on the memory-load tradeoff than the scheme in [28], while the needed subpacketizations of these two schemes are the same. Note that when $K > N + t$, the proposed demand-private scheme is also order-optimal within a constant gap, by using a similar proof as [28, Appendix D].[5]

Remark 2 (Comparison to the Virtual Users Scheme): A comparison on the loads of the virtual users scheme in Theorem 3 and our construction with the optimal PIR scheme in Corollary 1 is depicted in Figure 1. Note that the subpacketization of the virtual users scheme is $2^{\mathcal{H}(M/N)NK}$ and is at most exponential in NK, while that of Corollary 1 is $2^{\mathcal{H}(M/N)K}$ and is at most exponential in K.

Remark 3: The connection between PIR and demand private coded caching in our structure in Theorem 4, emerges from the fact that the individual queries sent to the servers solely do not reveal any information about the demanded file index. Therefore, the query to one server can be used to fill out the cache memory and the query to the other server to build up the server transmission, without revealing any information about the demanded indices by the users. This logic holds for any PIR scheme, including multi-message PIR schemes. In particular, if we assume $d_k, k \in [K]$ denotes the set of demands by user k, and $\mathbf{d} = (d_1, d_2, \ldots . d_K)$, then the proof of Theorem 4 in Appendix A works without any modifications. In this case by using these schemes, each user can request multiple files in the coded caching scheme while preserving the privacy of these demands.

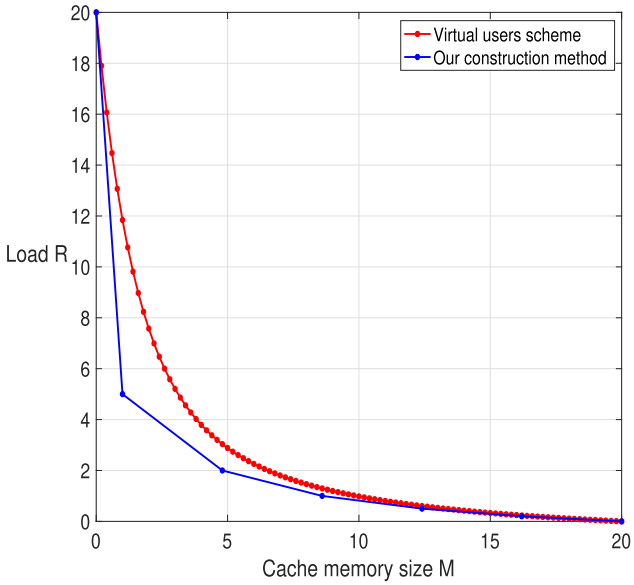

Fig. 1. Comparison of the loads for parameters $N = 20, K = 5$, and different values of M for the virtual users scheme in Theorem 3 and our construction in Corollary 1.

We can also extend the proposed construction in Theorem 4 to obtain a more flexible tradeoff between memory, load, and subpacketization, by using any coded caching scheme under PDA construction [13] instead of the MAN caching scheme (recall that the MAN scheme can also be seen as a caching scheme under PDA construction). This extension is feasible because the coded caching schemes under PDA construction are based on uncoded cache placement (which is symmetric across files) and clique-covering delivery.[6] Directly from the proof of Theorem 4, we can have the following corollary.

Corollary 2: Given any two-server PIR scheme with N files and download cost pair (R_{D_1}, R_{D_2}), and given any non-private coded caching scheme under PDA construction with memory-load tradeoff (M_1, R_1), there exists an (N, K) coded caching scheme with private demands which can achieve the memory-load tradeoff point

$$(M, R) = \big(M_1 + (1 - M_1/N)\left(\mu_1 R_{D_1} + \mu_2 R_{D_2}\right), \left(\mu_1 R_{D_2} + \mu_2 R_{D_1}\right) R_1\big), \tag{22}$$

where $\mu_1, \mu_2 \in [0, 1], \mu_1 + \mu_2 = 1$. Assume the subpacketizations of the given PIR scheme and of the non-private coded caching scheme are F' and F'', respectively; then the needed subpacketization of the resulting coded caching scheme with private demands is $F'F''$.

By applying coded caching schemes under PDA construction into Corollary 2, we can further reduce the subpacketization of the scheme in Theorem 4.

[5]More precisely, by the same proof for the case $M \leq 1$, we can show the load equal to N is order-optimal within a factor of 4; when $M > 1$, we can show that the gap between the proposed scheme and the MAN scheme is within a constant gap. In addition, the memory-sharing between $(0, N)$ and the MAN scheme is order-optimal within a factor of 4 [57]. So we can prove that our scheme is also order-optimal within a constant gap.

[6]The clique-covering delivery means that in the delivery phase, multiple multicast messages are broadcast to the users. Each multicast message is a sum of subfiles and is useful to a subset of users, where each user requests one subfile and caches all the other subfiles.

B. New Construction on Coded Caching With Private Demands and Caches

Next, we consider the construction of coded caching schemes with both demand privacy and cache privacy. This is given in the following result, proved in Appendix B.

Theorem 5: Given any two-server N-message PIR scheme satisfying the UDIQ condition in Definition 1 with download cost pair (R_{D_1}, R_{D_2}) where R_{D_i} corresponds to server i and time sharing parameters μ_1, μ_2 where $\mu_1, \mu_2 \in [0,1], \mu_1 + \mu_2 = 1$, there exists an (N, K) coded caching scheme with private demands and caches whose achieved memory-load tradeoff (M, R) is the lower convex envelope of $(0, N)$, $(N, 0)$, and the points in (19). Assume the needed subpacketization of the given PIR scheme is F', then the needed subpacketization for each point in (19) with $t \in [0 : K-1]$ is $\binom{K}{t}F'$.

The novelty of the construction in Theorem 4 is the generation of private keys by a two-server PIR scheme. In the privacy key scheme [28], in addition to caching subfiles as in the MAN caching scheme, for each set $\mathcal{V} \subseteq [K]$ where $k \notin \mathcal{V}$ and $|\mathcal{V}| = t$, each user k also caches a random linear combination of $W_{1,\mathcal{V}}, \ldots, W_{N,\mathcal{V}}$ (assumed to be $p_1 W_{1,\mathcal{V}} + \cdots + p_N W_{N,\mathcal{V}}$) in its caches as a private key, such that the effective demand of user k in the delivery phase becomes

$$p_1 W_{1,\mathcal{V}} + \cdots + p_{d_k-1} W_{d_k-1,\mathcal{V}} + (p_{d_k}+1) W_{d_k,\mathcal{V}} + p_{d_k+1} W_{d_k+1,\mathcal{V}} + \cdots + p_N W_{N,\mathcal{V}}.$$

In this way, the privacy of the user's demand is preserved. In our construction, instead of storing a random linear combination of $W_{1,\mathcal{V}}, \ldots, W_{N,\mathcal{V}}$, we apply an arbitrary two-server PIR scheme where we treat each of $W_{1,\mathcal{V}}, \ldots, W_{N,\mathcal{V}}$ as a file in the PIR problem. The answer of the first server in the PIR scheme serves as the private key stored by user k; according to the demand of user k, the answer of the second server in the PIR scheme serves as the effective request of user k. Then in Theorem 5, if the PIR scheme additionally satisfies the UDIQ condition, the resulting coded caching scheme satisfies the cache privacy condition in addition to the demand privacy condition.

From the same reasoning on deriving Corollary 2, we can also extend Theorem 5 by using other coded caching schemes under PDA construction, and obtain the following corollary.

Corollary 3: Given any two-server N-message PIR scheme satisfying the UDIQ condition in Definition 1 with download cost pair (R_{D_1}, R_{D_2}), and given any non-private coded caching scheme under PDA construction with memory-load tradeoff (M_1, R_1), there exists an (N, K) coded caching scheme with private demands and caches that achieves the memory-load tradeoff the point in (22). Assume the subpacketizations of the given PIR scheme and of the non-private coded caching scheme are F' and F'', respectively; then the needed subpacketization of the resulting coded caching scheme with private demands is $F'F''$.

C. New Construction on Two-Server PIR Schemes

By the proposed construction in Theorem 5 (resp. the one in Theorem 4), in order to design coded caching schemes with private demands and caches (resp. with private demands), our task is to design two-server PIR schemes under (resp. without) the UDIQ condition with total download cost and subpacketization level as low as possible. In the following we propose a new construction structure for two-server PIR schemes under the UDIQ condition by leveraging coded caching schemes. Intuitively, this idea stems from the observation that, the placement phase of coded caching does not reveal any information about the demands; and the observation that, given the transmission of the delivery phase, we can decode different files from different cache configurations. Hence, we can treat the cache configuration of a user as the transmission of one server in the PIR scheme and treat the delivery phase as the transmission of the other server in the PIR scheme. From the above explanation, we have the following construction.

Theorem 6 (From Coded Caching to PIR): Assume that there exists a coded caching scheme for N users and N files that achieves the memory-load trafeoff (M, R) with subpacketization F. Then there exists a two-server N-message PIR scheme satisfying the UDIQ condition in Definition 1 with the download cost pair $(R_{D_1}, R_{D_2}) = (M, R)$ and subpacketization F.

Proof: We consider the coded caching scheme for the shared link setting with N files and $K = N$ users. In the cache placement phase, each user $i \in [N]$ fills its cache with the content denoted by Z_i. In the delivery phase, each user requests a unique file. Thus, the demand vector $\mathbf{d} = (d_1, \ldots, d_N)$ is a permutation function $\pi(.)$ from $[N]$ to $[N]$. In the delivery phase, the server sends the message $X_{\mathbf{d}}$. Due to the decodability of the coded caching scheme, from $X_{\mathbf{d}}$ and Z_i, we can decode W_{d_i}, for each $i \in [N]$.

Next we use the above coded caching scheme to construct a two-server PIR scheme under the UDIQ condition. Let us go back to the PIR setting, where the user requests file W_θ where θ is distributed uniformly at random on $[N]$. The user generates a random variable r uniformly on $[N]$ and sends r as the query to the first server, in order to retrieve Z_r. In addition, to determine the demand vector, we first define $\mathbf{d}_c$ as $\mathbf{d}_c = (1, 2, \ldots, N)$. The demand vector $\mathbf{d}$ is determined as the cyclic shift of $\mathbf{d}_c$ by $< r - \theta >_N$ positions to the right; i.e. $\mathbf{d}(i) = \mathbf{d}_c(< i - < r - \theta >_N >_N)$.[7] Now the user sends $< r - \theta >_N$ as the query to the second server to retrieve $X_{\mathbf{d}}$.

Obviously, the query to the first server is independent of the demand. In addition, since r is generated independently and uniformly, the second server cannot get any information about θ. So the privacy constraint in PIR in (13) is satisfied. On the other hand, since $I(r; < r - \theta >_N | W_1, \ldots, W_N) = I(r; \theta) = 0$, the UDIQ condition in (1) is also satisfied by this scheme. □

We then apply the MAN coded scheme with memory-load tradeoff points $(M, R) = (t, \frac{N-t}{t+1})$ and subpacketization level $\binom{N}{t}$, for $t \in [0 : N]$, into the construction in Theorem 6.

Theorem 7: There exists a two-server PIR scheme satisfying the UDIQ condition in Definition 1, whose achieved download cost pair is the convex envelope of the points

[7] In this paper, we let $< \cdot >_a$ represent the modulo operation with integer quotient $a > 0$ and we let $< \cdot >_a \in \{1, \ldots, a\}$ (i.e., we let $< b >_a = a$ if a divides b).

$(R_{D_1}, R_{D_2}) = (t, \frac{N-t}{t+1})$ with subpacketization level $\binom{N}{t}$, for all $t \in [N]$. By letting $t = O\left(\sqrt{N}\right)$, the resulting two-server PIR scheme achieves the download costs R_{D_1} and R_{D_2} of order $O\left(\sqrt{N}\right)$ with subpacketization level $O\left(\sqrt{N}^{\sqrt{N}}\right)$ (considering highest order in the exponent).

Remark 4: In this paper we exploit the connection between PIR and coded caching, where we use one to build the other, as illustrated in Fig. 2. More precisely, in Theorem 5 (resp. Theorem 4) we propose a construction structure on demand and cache private (resp. demand private) caching schemes using two-server PIR schemes satisfying (resp. not satisfying) the UDIQ condition. Later in Theorem 6 we propose a construction structure on two-server PIR schemes that satisfy the UDIQ condition using coded caching.

Next, we derive a lower bound on the download costs of a two-server PIR scheme that satisfies the UDIQ condition in Definition 1 by using a cut-set argument. We assume that the sets of queries to Server 1 and Server 2 are $\mathcal{Q}_1$ and $\mathcal{Q}_2$, respectively. Consider the set of pairs of queries that based on the design of the PIR scheme can be sent in order to decode file W_τ, after receiving their corresponding answers; we denote this set by $\mathcal{U}_\tau$ as follows,

$$\mathcal{U}_\tau \triangleq \{(Q_1, Q_2) : Q_1 \in \mathcal{Q}_1, Q_2 \in \mathcal{Q}_2,\ (Q_1, Q_2) \text{ decodes } W_\tau\}. \tag{23}$$

For a particular choice of $q_1 \in \mathcal{Q}_1$, we define the set of all queries in the set $\mathcal{Q}_2$ that can together decode file W_τ as follows.

$$\mathcal{U}_{\tau|Q_1=q_1} \triangleq \{Q_2 : Q_2 \in \mathcal{Q}_2, (q_1, Q_2) \text{ decodes } W_\tau\}. \tag{24}$$

Similarly, we define

$$\mathcal{U}_{\tau|Q_2=q_2} \triangleq \{Q_1 : Q_1 \in \mathcal{Q}_1, (Q_1, q_2) \text{ decodes } W_\tau\}. \tag{25}$$

We propose the following converse bound, whose proof could be found in Appendix D.

Theorem 8: In a two-server PIR scheme that satisfies the UDIQ condition in Definition 1, denote the query sets to servers 1 and 2 respectively by $\mathcal{Q}_1$ and $\mathcal{Q}_2$, where $|\mathcal{Q}_1| = N_1$ and $|\mathcal{Q}_2| = N_2$; denote the download costs from servers 1 and 2 by R_{D_1} and R_{D_2}, respectively. If we have uniform query distribution for both servers; $\Pr(Q_1 = q_1 \in \mathcal{Q}_1) = 1/N_1$ and $\Pr(Q_2 = q_2 \in \mathcal{Q}_2) = 1/N_2$,[8] then,

1) for all $q_1 \in \mathcal{Q}_1$ and all $q_2 \in \mathcal{Q}_2$, we have

$$n_2 \triangleq \left|\mathcal{U}_{\tau|Q_1=q_1}\right|, n_1 \triangleq \left|\mathcal{U}_{\tau|Q_2=q_2}\right|,\ \forall \tau \in [N]; \tag{26}$$

2) we have

$$\frac{N_1}{n_1} \leq N, \frac{N_2}{n_2} \leq N; \tag{27}$$

3) we have

$$\min_{\substack{\alpha_1 \in [N_1], \alpha_2 \in [N_2], \\ \alpha_1 \alpha_2 = \left\lceil \frac{N_1}{n_1} \right\rceil = \left\lceil \frac{N_2}{n_2} \right\rceil}} \alpha_1 R_{D_1} + \alpha_2 R_{D_2} \geq N \tag{28}$$

[8]The two-server PIR schemes in Theorem 9, satisfy the uniform query distribution condition stated in Theorem 8. To the best of our knowledge, existing information-theoretic PIR schemes also satisfy this condition.

4) if we assume $R_{D_1} = R_{D_2} = R'_D$, we have

$$R'_D \geq \frac{N}{2\left(\sqrt{\left\lceil \frac{N_1}{n_1} \right\rceil} + 1\right)} = \frac{N}{2\left(\sqrt{\left\lceil \frac{N_2}{n_2} \right\rceil} + 1\right)} \geq \frac{N}{2\left(\sqrt{N} + 1\right)}. \tag{29}$$

Note that for any two-server PIR scheme, by using time-sharing we can always obtain another two-server PIR scheme with the same download costs from the two servers, where the total download cost is the same as the previous two-server PIR scheme. Hence, it can be seen from (29) that any two-server PIR scheme that satisfies the UDIQ condition in Definition 1 should have a total download cost

$$R_D \geq \frac{N}{\left(\sqrt{\left\lceil \frac{N_1}{n_1} \right\rceil} + 1\right)} = \frac{N}{\left(\sqrt{\left\lceil \frac{N_2}{n_2} \right\rceil} + 1\right)} \geq \frac{N}{\left(\sqrt{N} + 1\right)} = O\left(\sqrt{N}\right). \tag{30}$$

Comparing the converse bound in (30) and the proposed two-server PIR scheme in Theorem 7, we can obtain the following order-optimality result.

Corollary 4: The total download cost of the two-server PIR scheme in Theorem 7, which is equal to $O\left(\sqrt{N}\right)$, is order-optimal under the UDIQ constraint and uniform query.

For some special cases, more precisely for $N \in \{2, 3, 4\}$, in Appendix C we propose new two-server PIR schemes that satisfy the UDIQ condition, whose subpacketizations are lower and whose download costs are lower or equal compared to the two-server PIR scheme in Theorem 7.

Theorem 9: For the two-server PIR schemes that satisfy the UDIQ condition in Definition 1, 1) when $N = 2$, the download cost pair $(R_{D_1}, R_{D_2}) = (0.5, 1)$ (i.e., $R_D = 3/2$) is achievable and the required subpacketization is $F' = 1$;

2) when $N = 3$, the download cost pair $(R_{D_1}, R_{D_2}) = (1, 1)$ (i.e., $R_D = 2$) is achievable and the required subpacketization is $F' = 1$;

3) when $N = 4$, the download cost pair $(R_{D_1}, R_{D_2}) = (1, 1)$ (i.e, $R_D = 2$) is achievable and the required subpacketization is $F' = 1$.

Based on Theorems 9 and 8, we readily get the following result.

Corollary 5: The PIR schemes in Theorem 9 for the cases $N = 2$ and $N = 4$, meet the lower bound (28) in Theorem 8 with equality.

Proof: For the case $N = 2$, as we mention in Appendix C, we use the PIR scheme proposed in [47, Section III-A]. Note that this scheme has uniform distribution over queries. In this scheme $N_1 = N_2 = 2$, $n_1 = n_2 = 1$, and $R_{D_1} = 0.5, R_{D_2} = 1$. For the minimization in the left hand side of (28), we have $\alpha_1 \alpha_2 = \alpha' = 2$. The minimum happens when $\alpha_1 = 2, \alpha_2 = 1$. Then,

$$2R_{D_1} + R_{D_2} = 2 = N. \tag{31}$$

So this case holds (28) with equality.

For the case $N = 4$ introduced in Appendix C-B, we have $N_1 = N_2 = 4$, $n_1 = n_2 = 1$, and $R_{D_1} = 1, R_{D_2} = 1$. Again,

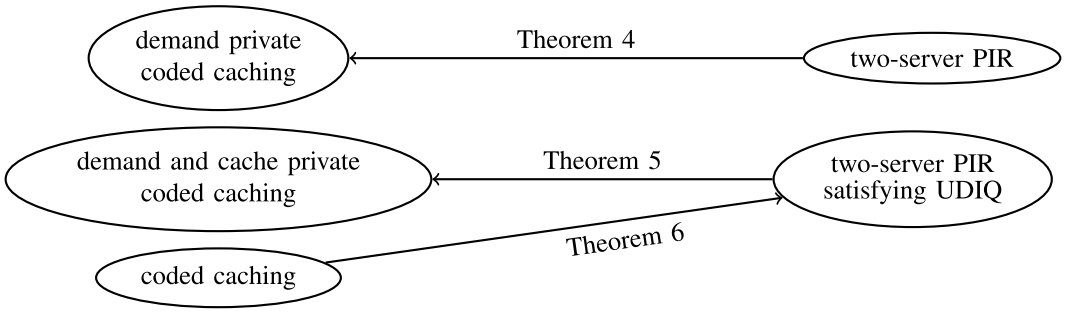

Fig. 2. Diagram of the proposed connections between PIR and coded caching.

remember that this scheme has a uniform distribution over queries. For the minimization on the left hand side of (28), we have $\alpha_1\alpha_2 = \alpha' = 4$. The minimum happens when $\alpha_1 = 2, \alpha_2 = 2$. Then,

$$2R_{D_1} + 2R_{D_2} = 4 = N. \tag{32}$$

So this case also holds (28) with equality. □

By applying the proposed two-server PIR schemes in Theorems 7 and 9 to our construction in Theorem 5, we can directly obtain the following coded caching schemes with private demands and caches. Note that for the first three parts, we use the schemes of Theorem 9, and for the last part, we use the scheme in Theorem 7.

Corollary 6: For the coded caching problem with private demands and caches, we have the following achievable schemes:

1) when $N = 2$, the following memory-load points are achievable,

$$(M, R) = \left(\frac{2t}{K} + (1 - \frac{t}{K})(\mu_1/2 + \mu_2), (\mu_1 + \mu_2/2)\frac{K-t}{t+1}\right), \quad \forall t \in [0 : K-1], \tag{33}$$

for $\mu_1 + \mu_2 = 1$ and $\mu_1, \mu_2 > 0$, while the required subpacketization is $\binom{K}{t}$;

2) when $N = 3$, the following memory-load points are achievable,

$$(M, R) = \left(\frac{3t}{K} + (1 - \frac{t}{K}), \frac{K-t}{t+1}\right), \forall t \in [0 : K-1], \tag{34}$$

while the required subpacketization is $\binom{K}{t}$;

3) when $N = 4$, the following memory-load points are achievable,

$$(M, R) = \left(\frac{4t}{K} + (1 - \frac{t}{K}), \frac{K-t}{t+1}\right), \forall t \in [0 : K-1], \tag{35}$$

while the required subpacketization is $\binom{K}{t}$;

4) when general N, the following memory-load points are achievable,

$$(M, R) = \left(\frac{t}{K}N + (1 - \frac{t}{K})O(\sqrt{N}), O(\sqrt{N})\frac{K-t}{t+1}\right), \quad \forall t \in [0 : K-1], \tag{36}$$

while the required subpacketization is $O\left(\binom{K}{t}\sqrt{N}^{\sqrt{N}}\right)$.

TABLE I
TWO-SERVER PIR SCHEME IN [47] FOR $N = K = 2$

	Server 1	Server 2	
		$d = 1$	$d = 2$
$T = 0$	0	W_1	W_2
$T = 1$	$W_1 + W_2$	W_2	W_1

For the general N, the proposed caching scheme with private demands and caches in (36) has a subpacketization level of $O\left(\binom{K}{t}\sqrt{N}^{\sqrt{N}}\right)$. Note that the subpacketization of the virtual users scheme in Theorem 3 is $\binom{NK}{t}$. Based on the asymptotic approximation of the binomial coefficients, the subpacketization of the virtual users scheme would be $F_1 = \binom{NK}{MK} \simeq 2^{NK\mathcal{H}(\frac{M}{N})}$, where $\mathcal{H}(.)$ is the binary entropy function. The subpacketization of our general scheme is of the order of $F_2 \simeq 2^{K\mathcal{H}(\frac{M}{N})}2^{\frac{1}{2}\sqrt{N}\log_2(N)}$. Then

$$\frac{F_2}{F_1} = \frac{2^{K\mathcal{H}(\frac{M}{N}) + \frac{1}{2}\sqrt{N}\log_2(N)}}{2^{NK\mathcal{H}(\frac{M}{N})}}. \tag{37}$$

If we assume $\frac{M}{N}$ is not vanishing with N, $\frac{F_2}{F_1}$ goes to 0 when N and K increase.

Remark 5: Based on Remark 1, the proposed demand and cache private coded caching schemes in Corollary 6 for $N \in \{2, 3, 4\}$, are optimal within a constant multiplicative factor.

At the end of this subsection, we illustrate the main idea of the construction in Theorem 4 with an example.

Example 1 ($K = N = 2, M = \frac{5}{4}$): In this example, we use the PIR scheme in [47, Section III-A], where the total download cost is $R_D = 3/2$ and the subpacketization level is $F' = 1$. Their scheme is presented in Table I.

Assume the two files are A and B. Each file is split into two equal-length and non-overlapping subfiles as $A = (A_1, A_2)$ and $B = (B_1, B_2)$.

Placement: For the first part of the cache, user 1 caches $Z_1 = (A_1, B_1)$ and user 2 caches $Z_2 = (A_2, B_2)$. As can be seen in the PIR scheme, $\mathcal{Q}_1 = \mathcal{Q}_2 = 2$. User i chooses $T_i \in \{0, 1\}$ each with probability $1/2$. Suppose $T_1 = 0$ and $T_2 = 1$. Based on our proposed approach in Theorem 5, the second user additionally caches $\gamma_1(Q_{1,2} = T_2 = 1, A_1, B_1) = A_1 + B_1$, while the first user caches nothing additional since $\gamma_1(Q_{1,1} = T_1 = 0, A_2, B_2) = 0$. So in total, the caches by the two users are

$$Z_1 = (A_1, B_1), \tag{38}$$
$$Z_2 = (A_2, B_2, A_1 + B_1). \tag{39}$$

TABLE II
DELIVERY PHASE OF DEMAND AND CACHE PRIVATE CODED CACHING SCHEME FOR $K = N = 2$ AND $M = \frac{5}{4}$

	$\mathbf{d} = (A, A)$	$\mathbf{d} = (A, B)$	$\mathbf{d} = (B, A)$	$\mathbf{d} = (B, B)$
$(T_1, T_2) = (0, 0)$	$A_2 + A_1$	$A_2 + B_1$	$B_2 + A_1$	$B_2 + B_1$
$(T_1, T_2) = (0, 1)$	$A_2 + B_1$	$A_2 + A_1$	$B_2 + B_1$	$B_2 + A_1$
$(T_1, T_2) = (1, 0)$	$B_2 + A_1$	$B_2 + B_1$	$A_2 + A_1$	$A_2 + B_1$
$(T_1, T_2) = (1, 1)$	$B_2 + B_1$	$B_2 + A_1$	$A_2 + B_1$	$A_2 + A_1$

Delivery: Assume that user 1 demands file A and user 2 demands file B. Since $\gamma_2(Q_{2,1} = T_1 = 0, A_2, B_2) = A_2$ and $\gamma_2(Q_{2,2} = T_2 = 1, A_1, B_1) = A_1$, the transmission of the server is $A_2 + A_1$. User 1 cancels out A_1 and recovers A_2. User 2 recovers B_1 by using the transmission $A_2 + A_1$ and the cached content A_2, $A_1 + B_1$. So both users receive their desired subfiles. For other cases of (T_1, T_2), the transmission of the server follows Table II. As can be seen, when $A_2 + A_1$ is sent by the server, there can be four different cases happening.

- $(T_1, T_2) = (0, 0)$ and demand vector $\mathbf{d} = (A, A)$;
- $(T_1, T_2) = (0, 1)$ and demand vector $\mathbf{d} = (A, B)$;
- $(T_1, T_2) = (1, 0)$ and demand vector $\mathbf{d} = (B, A)$;
- $(T_1, T_2) = (1, 1)$ and demand vector $\mathbf{d} = (B, B)$.

For user 1 who is aware of the values $T_1 = 0, d_1 = A$, there can exist two possible options of

- $(T_1, T_2) = (0, 0)$ and demand vector $\mathbf{d} = (A, A)$,
- $(T_1, T_2) = (0, 1)$ and demand vector $\mathbf{d} = (A, B)$,

which reveals no information about the value of neither d_2 nor T_2 since

$$\begin{aligned}
&\Pr(d_2 = A | T_1 = 0, d_1 = A, X_{\mathbf{d}} = A_2 + A_1) \\
&= \frac{\Pr(d_2 = A, T_1 = 0, d_1 = A, X_{\mathbf{d}} = A_2 + A_1)}{\Pr(T_1 = 0, d_1 = A, X_{\mathbf{d}} = A_2 + A_1)} \\
&= \frac{\Pr(d_2 = A, T_1 = 0, d_1 = A)}{\Pr(T_1 = 0, d_1 = A)} \\
&\times \frac{\Pr(X_{\mathbf{d}} = A_2 + A_1 | d_2 = A, T_1 = 0, d_1 = A)}{\Pr(X_{\mathbf{d}} = A_2 + A_1 | T_1 = 0, d_1 = A)} \\
&= \frac{(1/2)^3 (1/2)}{(1/2)^2 (1/2)} = \frac{1}{2},
\end{aligned} \tag{40}$$

and

$$\begin{aligned}
&\Pr(T_2 = 0 | T_1 = 0, d_1 = A, X_{\mathbf{d}} = A_2 + A_1) \\
&= \frac{\Pr(T_2 = 0, T_1 = 0, d_1 = A, X_{\mathbf{d}} = A_2 + A_1)}{\Pr(T_1 = 0, d_1 = A, X_{\mathbf{d}} = A_2 + A_1)} \\
&= \frac{\Pr(T_2 = 0, T_1 = 0, d_1 = A)}{\Pr(T_1 = 0, d_1 = A)} \\
&\times \frac{\Pr(X_{\mathbf{d}} = A_2 + A_1 | T_2 = 0, T_1 = 0, d_1 = A)}{\Pr(X_{\mathbf{d}} = A_2 + A_1 | T_1 = 0, d_1 = A)} \\
&= \frac{(1/2)^3 (1/2)}{(1/2)^2 (1/2)} = \frac{1}{2},
\end{aligned} \tag{41}$$

which equals the prior probability for d_2 and T_2. Thus, both the demand and the cache of user 2 is kept private. Similarly this holds for user 1.

Note that both the load $1/2$ and the cache size $5/4$ are expected values over the random choice of the queries to the first server in the placement phase and the corresponding queries to the second server in the delivery phase. Note that user 2 in this example has a cache size of $3/2$, but if it had chosen $T_2 = 0$ like the first user, it would have had a cache of size 1. So on average we have a cache size of $5/4$.

As a comparison, the privacy key scheme in [28] for the same system parameters of $K = N = 2, M = 5/4$ has a load of $R = 5/4$, while our scheme achieves the load $R = 1/2$ while additionally preserving cache privacy, which the privacy key scheme does not.

IV. CODED CACHING WITH PRIVATE DEMANDS AND IMPERFECTLY PRIVATE CACHES

Since constructing two-server PIR schemes under the UDIQ property is difficult, and in any case the download cost R_D increases at least as $O(\sqrt{N})$ (see Theorem 8), to be able to propose better PIR schemes in terms of download cost, which leads to better memory-load tradeoffs for the corresponding caching scheme (see Theorem 4), in this section we relax the perfect cache privacy and allow some leakage on the cache information, while preserving perfect demand privacy.

We first review the leakage metric in the leaky PIR literature, and then introduce our leakage metric. Next, we apply the two-server PIR scheme in [43] to our construction structure in Theorem 4, and compute the cache leakage of the resulting demand private coded caching scheme. Finally, we compare the resulting schemes with the existing demand private coded caching schemes, in terms of load and cache leakage.

A. Cache Information Leakage

Privacy leakage has already been introduced in several works on PIR following various definitions (see [58], [59], [60], [61], [62], [63], [64]). In this section we introduce a privacy leakage definition on the cache information that is relevant to our setting. The decoding and demand privacy constraints remain the same as in (5) and (6), while the cache privacy constraint in (7) no longer exists. As the cache privacy constraint in (7) suggests, the perfect scenario for the cache memory is that the ambiguity of its information does not change conditioned on the knowledge of the server transmission. In a non-perfect scenario, we want to keep the distribution on cache information before and after server transmission close to each other as much as possible.

In information-theoretic secrecy [65], the *information leakage rate* associated with the $(2^{nR}, n)$ secrecy code is defined as

$$\frac{1}{n} I(M, Z^n), \tag{42}$$

where M represents the sender's message and Z^n represents the message received by the eavesdropper for the block length n. In our definition for cache privacy, the server's transmission $X_{\mathbf{d}}$ acts as the message received by the eavesdropper, and user k's cache metadata $\mathscr{M}_k$ acts as the message we want to keep private. We replace the block length n with the entropy of the cache metadata as the block length for the message. This motivates our consideration of the following cache leakage metric for user k:

$$\epsilon_k = \frac{I(\mathscr{M}_k; X_{\mathbf{d}})}{H(\mathscr{M}_k)} = 1 - \frac{H(\mathscr{M}_k | X_{\mathbf{d}})}{H(\mathscr{M}_k)}, \ \forall k \in [K] \tag{43}$$

where $H(\cdot)$ is the entropy function. In the fully private case when there is no leakage, this metric is 0. As the uncertainty amount on cache information decreases after server transmission, the leakage increases and goes to 1 when the cache information is fully leaked.

B. Cache-Leakages of [26] and [28]

We then consider the coded caching schemes with private demands in [26] and [28], and compute their cache leakage. In the case of single file requests in [28], the randomness on the cache for user k is $\mathscr{M}_k = \mathbf{p}_k := (p_{k,1}, \ldots, p_{k,N})$, chosen uniformly at random from $\mathbb{F}_q^k$, such that the summation of the elements of $\mathbf{p}_k$ equals $q-1$; $\sum_{n\in[N]} p_{k,n} = q-1$. Because of this constraint, the total number of choices for $\mathbf{p}_k$ is q^{N-1}. So we have

$$H(\mathscr{M}_k) = (N-1)\log(q). \tag{44}$$

If we denote the demand vector for user k by $\mathbf{d}_k$ which for single file demands has a 1 at the position of requested file index and 0 elsewhere, the server sends $\mathbf{q}_k = \mathbf{p}_k + \mathbf{d}_k$ for all $k \in [K]$ as metadata alongside the main message. Having $\mathbf{q}_k$, since there are only N options for $\mathbf{d}_k$ (uniformly chosen), our options for $\mathbf{p}_k$ would also be limited to N. Thus

$$H(\mathscr{M}_k | X_{\mathbf{d}}) = \log(N). \tag{45}$$

According to (43) we have

$$\epsilon_k = 1 - \frac{1}{\log(q)}\frac{\log(N)}{(N-1)}, \tag{46}$$

which goes to 1 as N increases. Our goal is to introduce a coded caching scheme with non-zero leakage on cache using a two-server PIR scheme that does not satisfy the UDIQ condition in Definition 1, instead of the perfectly private scheme of Theorem 7, with the benefit of achieving better download costs and subpacketization for the PIR scheme, which will directly affect the memory-load tradeoff and subpacketization of the resulting coded caching scheme based on our structure in Theorem 5.

For the virtual users scheme of [26], the cache of user k is selected among N choices uniformly at random. After the transmission, the probabilty distribution over cache information does not change as proved in Theorem 3. So in this case, the leakage would be $\epsilon_k = 0$ for all users, which is perfect, but as mentioned before, this scheme has a huge subpacketization level.

C. Review on [42] and [43]

We then review the protocol proposed in [43] with the lowest communication cost (equal to $N^{o(1)}$) among all existing two-server PIR protocols, which will be applied in our proposed construction structure in Theorem 4. This scheme is a combination of an existing two-server PIR scheme which uses polynomial interpolation [42] and Matching Vector Codes (MV codes) [44], [45]. We will briefly go through [42] and then introduce matching vector families and then describe the protocol in [43].

The scheme in [42] is based on building polynomials of degree 3. First, choose k such that $N \leq \binom{k}{3}$. Pick a finite field $\mathbb{F}_q$ where $q > 3$. Define an encoding ϕ that maps indices in $[N]$ to binary k-dimensional space.

$$\phi : [N] \to \{0,1\}^k \subset \mathbb{F}_q^k, \tag{47}$$

such that the resulting k-dimensional codewords are of Hamming weight 3. If we denote the k-dimensional space by $\mathbf{x} = (x_1, \ldots, x_k)$, the polynomial $F(\mathbf{x}) \in \mathbb{F}_q[x_1, \ldots, x_k]$ where $\mathbb{F}_q[x_1, \ldots, x_k]$ denotes the field of polynomials with the variables $x_1, \ldots, x_k$ over $\mathbb{F}_q$, is defined as follows,

$$F(\mathbf{x}) = \sum_{i=1}^{N} W_i \left(\prod_{j:\phi(i)_j=1} x_j \right), \tag{48}$$

where the files W_i are considered to be one bit. This polynomial satisfies $F(\phi(i)) = W_i, \forall i \in [N]$.

Suppose the user demands the file W_τ. The scheme works as follows:

- the user picks a $\mathbf{z} \in \mathbb{F}_q^k$ uniformly at random;
- the user sends $\phi(\tau) + t_i\mathbf{z}$ to server i where $t_1 \neq t_2 \in \mathbb{F}_q \backslash \{0\}$;
- server i sends to the user the values $F(\phi(\tau) + t_i\mathbf{z})$ and $\nabla F(\phi(\tau) + t_i\mathbf{z})$.

With the answers received from both servers, the user can retrieve $F(\phi(\tau)) = W_\tau$; the reader can refer to [42] for the detailed proof of decodability. The privacy of demand is protected since $\phi(\tau) + t_i\mathbf{z}$ is uniformly distributed in $\mathbb{F}_q^k$ for any value of τ.

We then review the two-server PIR scheme in [43], starting with the following definition.

Definition 2 (Matching Vector Family): Let $S \subset \mathbb{Z}_m \backslash \{0\}$ and let $\mathcal{F} = (\mathcal{U}, \mathcal{V})$ where $\mathcal{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_N)$, $\mathcal{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_N)$, and $\mathbf{u}_i, \mathbf{v}_i \in \mathbb{Z}_m^w, \forall i \in [N]$. Then $\mathcal{F}$ is called an S-matching vector family of size N and dimension w if $\forall i, j$,

$$\langle \mathbf{u}_i, \mathbf{v}_j \rangle \begin{cases} = 0 & \text{if } i = j \\ \in S & \text{if } i \neq j, \end{cases} \tag{49}$$

where $\langle \mathbf{u}_i, \mathbf{v}_j \rangle$ indicates the inner product between the two vectors. It has been shown that based on [66, Theorem 1.2], for $S = \{1, 3, 4\}$, we can build matching vector codes with parameters N and w (and when m is composite) such that

$$w = \exp\left(O\left(\sqrt{\log N \log\log N} \right) \right). \tag{50}$$

For a commutative ring $\mathcal{R}$, the ring of polynomials in variables $x_1, \ldots, x_w$ with coefficients in $\mathcal{R}$ is denoted by $\mathcal{R}[x_1, \ldots, x_w]$. In [43], the authors introduce a definition to extend the notion of partial derivatives to polynomials in $\mathcal{R}[x_1, \ldots, x_w]$ as follows.

Definition 3: Let $\mathcal{R}$ be a commutative ring and let $F(\mathbf{x}) = \sum c_{\mathbf{z}} \mathbf{x}^{\mathbf{z}} \in \mathcal{R}[x_1, \ldots, x_w]$. We define $F^{(1)}(\mathbf{x}) \in (\mathcal{R}^w)[x_1, \ldots, x_w]$ to be

$$F^{(1)}(\mathbf{x}) := \sum (c_{\mathbf{z}} \cdot \mathbf{z}) \mathbf{x}^{\mathbf{z}}, \tag{51}$$

where $\mathbf{x}^{\mathbf{z}} = x_1^{z_1} x_2^{z_2} \ldots x_w^{z_w}$. Now we are ready to introduce the scheme in [43].

For the rest of this section, $\mathcal{R} = \mathcal{R}_{6,6} = \mathbb{Z}_6[\gamma]/(\gamma^6 - 1)$ which is the ring of univariate polynomials $\mathbb{Z}_6[\gamma]$ modulo the identity $\gamma^6 = 1$ as defined in [43]. It should be noted that the set S which contains only three values is the key to this scheme since, roughly speaking, the powers of γ appearing in the polynomial are from this set and 0 and therefore, there will be four unknown coefficients and we would only need two evaluations and two derivatives to recover the intended value. We will not go into the details of the recovery and refer the reader to the paper.

Assume the user's demand is W_τ. The servers save the data in the polynomial $F(\mathbf{x}) \in \mathcal{R}[x_1, \ldots, x_w]$ where

$$F(\mathbf{x}) = F(x_1, \ldots, x_w) = \sum_{i=1}^{N} W_i \mathbf{x}^{\mathbf{u}_i}, \tag{52}$$

where $\mathcal{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_N)$ is given by the matching vector family $\mathcal{F} = (\mathcal{U}, \mathcal{V})$ for $m = 6$ and w as in (50). Then,

- the user picks a $\mathbf{z} \in \mathbb{Z}_6^w$ uniformly at random;
- the user sends $\mathbf{z} + t_i \mathbf{v}_\tau$ to server i;
- server i sends back the values $F(\gamma^{\mathbf{z}+t_i\mathbf{v}_\tau})$ and $F^{(1)}(\gamma^{\mathbf{z}+t_i\mathbf{v}_\tau})$,

where the vector $(\gamma^{z_1+t_i\mathbf{v}_{\tau,1}}, \gamma^{z_2+t_i\mathbf{v}_{\tau,2}}, \ldots, \gamma^{z_w+t_i\mathbf{v}_{\tau,w}})$ is denoted by $\gamma^{\mathbf{z}+t_i\mathbf{v}_\tau}$. Since the values $\mathbf{z} + t_i\mathbf{v}_\tau$ are distributed uniformly on $\mathbb{Z}_6^w$, the privacy of demand in the PIR scheme is preserved. Also in the scheme $t_1 = 0$ and $t_2 = 1$. Since the user sends elements in $\mathbb{Z}_6^w$ to both servers and receives an element in $\mathcal{R}$ and another one in $\mathcal{R}^w$ from each server, the communication cost would be $O(w) = N^{O\left(\sqrt{\frac{\log\log N}{\log N}}\right)}$.

D. Coded Caching Schemes With Private Demands and Imperfectly Private Caches Based on Theorem 4

We now apply the two-server PIR scheme in [43] to our structure in Theorem 4. Assume we have a system of K users and N files $W_1, \ldots, W_N$. The server is connected to the users with a shared link. For any $t = KM/N \in [K]$, each file is split into $\binom{K}{t}$ non-overlapping subfiles of the same size,

$$W_n = (W_{n,\tau} : \tau \subset [K], |\tau| = t). \tag{53}$$

We assume that each subfile has one bit; but we can easily extend the scheme for the other case.

Placement phase: In the first part of the placement phase, for each $k \in [K]$, any subfile $W_{n,\tau}$ with $k \in \tau$ is stored in the cache. Therefore,

$$\{W_{n,\tau} : n \in [N], \tau \subset [K], |\tau| = t, k \in \tau\} \subset Z_k. \tag{54}$$

In the second part of the placement phase, for each set $\tau \subseteq [K]$ where $|\tau| = t$ and $k \notin \tau$, user k caches the result of an encoding on all subfiles $\{W_{n,\tau}, n \in [N]\}$. The matching vector family $\mathcal{F} = (\mathcal{U}, \mathcal{V})$ is constructed where $\mathcal{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_N)$ and $\mathcal{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_N)$ such that $\mathbf{u}_i, \mathbf{v}_i \in \mathbb{Z}_6^w, \forall i \in [N]$ as explained previously.

User k picks $\mathbf{z}_k \in \mathbb{Z}_6^w$ uniformly at random. The user sends $\mathbf{z}_k$ to the server. For each τ such that $k \notin \tau$, the server sends $F(\gamma^{\mathbf{z}_k}, W_{[N],\tau})$ and $F^{(1)}(\gamma^{\mathbf{z}_k}, W_{[N],\tau})$ to the user where

$$F(\mathbf{x}, W_{[N],\tau}) = F(x_1, \ldots, x_k, W_{1,\tau}, \ldots, W_{N,\tau})$$
$$= \sum_{i=1}^{N} W_{i,\tau} \mathbf{x}^{\mathbf{u}_i}. \tag{55}$$

This completes the placement phase.

Delivery phase: Assume that user k demands the file W_{τ_k}. In the delivery phase, user k sends $\mathbf{z}_k + \mathbf{v}_{\tau_k}$ to the server. For each $\mathcal{S} \subset [K]$, where $|\mathcal{S}| = t + 1$, the server sends the multicast messages

$$Y_{\mathcal{S}} = \left(\sum_{s\in\mathcal{S}} F\left(\gamma^{\mathbf{z}_s+\mathbf{v}_{\tau_s}}, W_{[N],\mathcal{S}\setminus s}\right), \sum_{s\in\mathcal{S}} F^{(1)}\left(\gamma^{\mathbf{z}_s+\mathbf{v}_{\tau_s}}, W_{[N],\mathcal{S}\setminus s}\right) \right). \tag{56}$$

Along with the messages $Y_{\mathcal{S}}$, in order for the users to be able to decode their needed messages, the server should send the values $\{\mathbf{z}_k + \mathbf{v}_{\tau_k}, \forall k \in [K]\}$ as metadata. So the transmitted message by the server $X_{\mathbf{d}}$ would be

$$X_{\mathbf{d}} = \{Y_{\mathcal{S}}, \mathcal{S} \subseteq [K], |\mathcal{S}| = t + 1\} \bigcup \{\mathbf{z}_k + \mathbf{v}_{\tau_k}, \forall k \in [K]\}. \tag{57}$$

The decodability proof follows from the proof of Theorem 5.

Performance: An observation on the scheme reveals that the cache metadata equals $\mathscr{M}_k = \mathbf{z}_k$. Now we compute the amount of cache information leakage based on (43). Since $\mathscr{M}_k$ is uniformly distributed in $\mathbb{Z}_6^w$,

$$H(\mathscr{M}_k) = w \log 6. \tag{58}$$

In addition, we have

$$H(\mathscr{M}_k | X_{\mathbf{d}}) = H\left(\mathbf{z}_k | \{\mathbf{z}_{k'} + \mathbf{v}_{\tau_{k'}}, \forall k' \in [K]\}\right) \tag{59a}$$
$$= H\left(\mathbf{z}_k | \mathbf{z}_k + \mathbf{v}_{\tau_k}\right) \tag{59b}$$
$$= \log N, \tag{59c}$$

where (59a) comes from the fact that the values $Y_{\mathcal{S}}$ depend on $\{\mathbf{z}_{k'} + \mathbf{v}_{\tau_{k'}}, \forall k' \in [K]\}$ and the library and (59b) comes from independence of $\mathbf{z}_k$ values for all $k \in [K]$. Therefore

$$\epsilon_k = 1 - \frac{\log N}{w \log 6} = 1 - O\left(\frac{\log N}{N^{\sqrt{\frac{\log\log N}{\log N}}}}\right). \tag{60}$$

Since the total download cost of this scheme is $O(w) = N^{O\left(\sqrt{\frac{\log\log N}{\log N}}\right)}$, based on our structure in Theorem 5, we can achieve the lower convex envelope of the memory-load pair points

$$(M, R) = \left(\frac{Nt}{K} + \left(1 - \frac{t}{K}\right) N^{O\left(\sqrt{\frac{\log\log N}{\log N}}\right)}, N^{O\left(\sqrt{\frac{\log\log N}{\log N}}\right)} \frac{K - t}{t + 1}\right). \tag{61}$$

To compare, the memory-load tradeoff and cache leakage of the privacy key scheme of [28] follows $\left(1 + \frac{t(N-1)}{K}, \frac{\binom{K}{t+1} - \binom{K-\min\{N-1,K\}}{t+1}}{\binom{K}{t}}\right)$, $\forall t \in [0 : K]$ and $\epsilon_k = 1 - \frac{1}{\log(q)} \frac{\log(N)}{(N-1)}$ respectively, while for our scheme for general N in Corollary 6, we have $\left(\frac{t}{K}N + (1 - \frac{t}{K})O(\sqrt{N}), O(\sqrt{N})\frac{K-t}{t+1}\right), \forall t \in [0 : K - 1]$

as memory-load pair and $\epsilon_k = 0$. For the scheme in this section these parameters are $\left(\frac{Nt}{K} + \left(1-\frac{t}{K}\right) N^{O\left(\sqrt{\frac{\log\log N}{\log N}}\right)}, N^{O\left(\sqrt{\frac{\log\log N}{\log N}}\right)} \frac{K-t}{t+1}\right), \forall t \in [0:K-1]$ and $\epsilon_k = 1 - O\left(\frac{\log N}{N^{\sqrt{\frac{\log\log N}{\log N}}}}\right)$. The scheme in this section performs better in terms of cache leakage than the privacy key scheme since it converges much more slowly to 1, but has worse load. On the other hand, compared to our perfectly private scheme, it has a better load but of course worse leakage on cache.

At the end of this section, we provide an example to illustrate the proposed coded caching scheme with private demands and imperfectly private caches by leveraging the two-server PIR scheme in [43].

Example 2 ($N = K = 2$ and $t = 1$): This is an example just to demonstrate the placement and delivery phases of the proposed scheme. Therefore, we will not care about the S and w parameters of the matching vector family. In this scheme, $S = \{1\}$ and $w = 2$. We consider the coded caching problem with $N = K = 2$ and $t = KM/N = 1$. Each file is split into $\binom{K}{t} = 2$ non-overlapping equally-sized subfiles, i.e. $A = (A_1, A_2), B = (B_1, B_2)$. In the first part of the placement phase, each user's cache will be as follows,

$$Z_1 = (A_1, B_1), \tag{62}$$
$$Z_2 = (A_2, B_2). \tag{63}$$

For the second part of the placement, we first introduce a matching vector family based on Definition 2. We define the 2-tuples $\mathcal{U}$ and $\mathcal{V}$ of the matching vector family as follows,

$$\mathcal{U} = ((0,1),(1,0)), \tag{64}$$
$$\mathcal{V} = ((1,0),(0,1)). \tag{65}$$

The polynomial $F(\mathbf{x})$ in (52) for library files W_1 and W_2 is as follows,

$$F(\mathbf{x}) = W_1 x_2 + W_2 x_1 \tag{66}$$

In addition, the function $F^{(1)}(\mathbf{x})$ in (51) would be,

$$F^{(1)}(\mathbf{x}) = (W_2 x_1, W_1 x_2) \tag{67}$$

For the second part of the placement phase, user k chooses $\mathbf{z}_k \in \mathbb{Z}_6^2$ uniformly at random. Suppose the choices are $\mathbf{z}_1 = (2,3), \mathbf{z}_2 = (5,1)$. Users send these values to the server. The server sends back the pair $\left(F(\gamma^{\mathbf{z}_1}), F^{(1)}(\gamma^{\mathbf{z}_1})\right)$ to user 1 when $W_1 = A_2, W_2 = B_2$ and $\left(F(\gamma^{\mathbf{z}_2}), F^{(1)}(\gamma^{\mathbf{z}_2})\right)$ to user 2 when $W_1 = A_1, W_2 = B_1$. So in total the caches are as follows,

$$Z_1 = (A_1, B_1, A_2\gamma^3 + B_2\gamma^2, (B_2\gamma^2, A_2\gamma^3)), \tag{68}$$
$$Z_2 = (A_2, B_2, A_1\gamma + B_1\gamma^5, (B_1\gamma^5, A_1\gamma)). \tag{69}$$

In the delivery phase, suppose the demand for users 1 and 2 are A, B, respectively. When the server receives the demands, it should compute for user 1 the values $\left(F(\gamma^{\mathbf{z}_1+\mathbf{v}_1}), F^{(1)}(\gamma^{\mathbf{z}_1+\mathbf{v}_1})\right)$ when $W_1 = A_2, W_2 = B_2$ and for user 2 the values $\left(F(\gamma^{\mathbf{z}_2+\mathbf{v}_2}), F^{(1)}(\gamma^{\mathbf{z}_2+\mathbf{v}_2})\right)$ when $W_1 = A_1, W_2 = B_1$. Then adds each part together and sends the multicast messages

$$(A_2\gamma^3 + B_2\gamma^3) + (A_1\gamma^2 + B_1\gamma^5), \tag{70}$$

$$\left(B_2\gamma^3 + B_1\gamma^5, A_2\gamma^3 + A_1\gamma^2\right), \tag{71}$$

including the metadata to the users on the shared channel. Using this transmission and its cache content, user 1 recovers $A_2\gamma^3 + B_2\gamma^3$ and $(B_2\gamma^3, A_2\gamma^3)$ and user 2 recovers $A_1\gamma^2 + B_1\gamma^5$ and $(B_1\gamma^5, A_1\gamma^2)$ and using the decoding procedure for the PIR scheme, each user can decode its demanded file. The privacy of demands is fully satisfied; this is because, from the metadata $\mathbf{z}_2 + \mathbf{v}_2 = (5,2)$, user 1 would not know any information about the value $\mathbf{v}_2$ since $\mathbf{z}_2$ is uniformly distributed on $\mathbb{Z}_6^2$. On the other hand, the cache is not perfectly private. The cache leakage in this example equals $\epsilon_k = 1 - \frac{\log N}{w \log 6} = 1 - \frac{\log 2}{2 \log 6}$. □

V. Conclusion

In this paper, we formulated the coded caching problem with private demands and caches, where we added the privacy constraint on users' caches to the existing coded caching problem with private demands. We first showed that the existing demand-private coded caching scheme, which is based on the introduction of virtual users, can also preserve the privacy of caches while suffering from a super high subpacketization. The main contribution of this paper was to propose a new structure for constructing demand- and cache-private coded caching schemes by using two-server PIR schemes with uniform demand and independent queries. Using this structure and the newly designed PIR schemes, we were able to propose demand and cache private coded caching schemes with significant reduction on the subpacketization compared to the virtual users scheme. As a by-product, our structure is also able to design order-optimal demand private coded caching schemes, which demonstrates its flexibility. In designing the two-server PIR schemes that satisfy the aforementioned constraint, we have introduced a novel structure to leverage from coded caching. These two structures close the loop in the connection between coded caching and PIR in this paper. Furthermore, we propose a converse bound on the download costs of this particular class of PIR schemes which reveals the order-optimality of the designed achievable scheme. We then extended the proposed structure to the coded caching problem with private demands and imperfectly private caches. Future and ongoing works include providing a lower bound on the memory-load tradeoff of demand and cache private caching schemes, designing two-server PIR schemes for general N with less subvpacketization compared to the proposed one, studying the tradeoff between the amount of leakage and system parameters of the PIR scheme in the imperfect private caches scenario.

Appendix A
Proof of Theorem 4: New Construction on Coded Caching Schemes With Demand Privacy

We assume that the set of queries sent to servers 1 and 2 in the PIR scheme are chosen from the sets $\mathcal{Q}_1$ and $\mathcal{Q}_2$ respectively. Note that if a PIR scheme with download costs R_{D_1} and R_{D_2} corresponding to servers 1 and 2 is achievable, then by a time-sharing argument, the download cost pair

$(R'_{D_1}, R'_{D_2}) = (\mu_1 R_{D_1} + \mu_2 R_{D_2}, \mu_1 R_{D_2} + \mu_2 R_{D_1})$ where $\mu_1, \mu_2 \in [0,1], \mu_1 + \mu_2 + 1$ is also achievable.

Placement: The placement phase is divided into two steps. The first step follows exactly the same procedure as the MAN placement phase. For each $t \in [0 : K-1]$, each file is split into $\binom{K}{t}$ non-overlapping subfiles of the same size,

$$W_n = (W_{n,\tau} : \tau \subset [K], |\tau| = t), \tag{72}$$

where each subfile contains $F/\binom{K}{t}$ bits. In addition, for each $n \in [N]$ and each $\tau \subset [K]$ where $|\tau| = t$, we divide each $W_{n,\tau}$ into F' non-overlapping subfiles $W_{n,\tau,\omega}$ of the same size,

$$W_{n,\tau} = \{W_{n,\tau,\omega}, \omega \in [F']\}, \tag{73}$$

where we recall that F' represents the subpacketization of the two-server PIR scheme.

Each user $k \in [K]$ first caches $W_{n,\tau}$ where $k \in \tau$; in other words,

$$\{W_{n,\tau} : n \in [N], \tau \subset [K], |\tau| = t, k \in \tau\} \subset Z_k. \tag{74}$$

In the second step of the placement phase, for each index τ where $k \notin \tau$, user k caches an encoding function on all subfiles $\{W_{n,\tau}, n \in [N]\}$. The encoding is chosen as follows. First, user k chooses a query $Q_{1,k}$ from $\mathcal{Q}_1$ uniformly at random. Then the encoding function would be the answer of the first server in the PIR scheme when the query is $Q_{1,k}$ and the files are $W_{1,\tau}, \ldots, W_{N,\tau}$, i.e. $\gamma_1(Q_{1,k}, W_{1,\tau}, \ldots, W_{N,\tau})$. The second part of file splitting in (73) is necessary to compute this encoding function. Thus, the second part of the cache for user k would be

$$\{\gamma_1(Q_{1,k}, W_{1,\tau}, \ldots, W_{N,\tau}), \tau \subset [K], |\tau| = t, k \notin \tau\} \subset Z_k. \tag{75}$$

Therefore in total, for every $k \in [K]$, Z_k would be

$$\begin{aligned} Z_k = &\{W_{n,\tau} : n \in [N], \tau \subset [K], |\tau| = t, k \in \tau\} \\ &\bigcup \{\gamma_1(Q_{1,k}, W_{1,\tau}, \ldots, W_{N,\tau}), \tau \subset [K], |\tau| = t, k \notin \tau\}. \end{aligned} \tag{76}$$

totally containing $\frac{N\binom{K-1}{t-1}}{\binom{K}{t}} F + R'_{D_1} \frac{\binom{K-1}{t}}{\binom{K}{t}} F = \left(\frac{Nt}{K} + R'_{D_1} \frac{K-t}{K}\right) F = MF$, satisfying the memory size constraint.

Delivery: Recall that for a (N, K) MAN coded caching scheme, for each $\mathcal{S} \subset [K]$ such that $|\mathcal{S}| = t+1$, the server transmits

$$\oplus_{s \in \mathcal{S}} W_{d_s, \mathcal{S}\setminus\{s\}}, \tag{77}$$

where $\oplus$ stands for bitwise XOR. Instead, in the delivery phase of our scheme, for each subset $\mathcal{S} \subseteq [K]$ where $|\mathcal{S}| = t+1$, the server transmits a multicast message as

$$Y_{\mathcal{S}} = \Sigma_{s \in \mathcal{S}} \gamma_2\big(Q_{2,s}, W_{1,\mathcal{S}\setminus\{s\}}, \ldots, W_{N,\mathcal{S}\setminus\{s\}}\big). \tag{78}$$

where $\gamma_2(.)$ is the answer encoding function of server 2 of the PIR scheme and $Q_{2,s}$ is chosen is such a way that the query pair $(Q_{1,s}, Q_{2,s})$ where $Q_{1,s}$ was chosen in the placement phase, corresponds to the d_s^{th} message in the PIR problem. This means that for every τ such that $k \notin \tau$, the answer of server 1, $\gamma_1\big(Q_{1,k}, W_{1,\tau}, \ldots, W_{N,\tau}\big)$, saved in the cache and the answer of server 2, $\gamma_2\big(Q_{2,k}, W_{1,\tau}, \ldots, W_{N,\tau}\big)$, extracted from the message $Y_{\mathcal{S}}$ with $\mathcal{S} = \tau \cup \{k\}$, lead user k to decode subfile $W_{d_k,\tau}$. Following the same procedure for all needed subfiles, user k decodes file W_{d_k}. This proves the satisfaction of the decodability condition in (5). It can be seen that in the delivery phase the server in total transmits $R'_{D_2} \frac{\binom{K}{t+1}}{\binom{K}{t}} F = R'_{D_2} \frac{K-t}{t+1} F$, coinciding with (19).

We should note that along with the multicast messages, the server should send also the values $\{Q_{2,k}, k \in [K]\}$ as metadata so that everyone can decode their required messages. We assume that the size of this metadata is negligible compared to the size of the multicast messages. Thus, the transmitted message $X_{\mathbf{d}}$ will be as follows,

$$X_{\mathbf{d}} = \{Y_{\mathcal{S}}, \mathcal{S} \subseteq [K], |\mathcal{S}| = t+1\} \bigcup \{Q_{2,k}, k \in [K]\} \tag{79}$$

Now we can check the demand privacy condition in (6).

$$\begin{aligned} &I(\mathbf{d}; X_{\mathbf{d}} | d_k, Z_k) \\ &\le I(\mathbf{d}; X_{\mathbf{d}} | d_k, Z_k, W_{[N]}) && \text{(80a)} \\ &\le I(\mathbf{d}; \{Q_{2,k}, k \in [K]\} | d_k, Z_k, W_{[N]}) && \text{(80b)} \\ &= \sum_{k' \in [K]\setminus\{k\}} I(d_{k'}; Q_{2,k'} | W_{[N]}) = 0 && \text{(80c)} \end{aligned}$$

where (80a) follows from (4), (80b) comes from the fact that the set $\{Y_{\mathcal{S}}, \mathcal{S} \subseteq [K], |\mathcal{S}| = t+1\}$ is a function of $\{Q_{2,k}, k \in [K]\}$ and $W_{[N]}$, (80c) follows from the fact that the pairs $(d_i, Q_{2,i})$ where $i \in [K]$ are independent of each other given $W_{[N]}$ by our construction and the privacy constraint in (13).

Appendix B
Proof of Theorem 5

Based on the proof of Theorem 4, we proved that our construction satisfies the decodability and demand privacy conditions in (5) and (6), respectively for any two-server PIR scheme. In this section, for PIR schemes that satisfy the UDIQ condition in Definition 1 additionally, we only need to prove that the privacy condition in (7) holds. For the cache privacy constraint in (7), for each $k \in [K]$ we have

$$\begin{aligned} &I\big((\mathscr{M}_1, \ldots, \mathscr{M}_K); X_{\mathbf{d}} | d_k, Z_k\big) \\ &\le I\big((\mathscr{M}_1, \ldots, \mathscr{M}_K); X_{\mathbf{d}} | d_k, Z_k, W_{[N]}\big) && \text{(81a)} \\ &\le I\big(Q_{1,1}, \ldots, Q_{1,K}; Q_{2,1}, \ldots, Q_{2,K} | d_k, Z_k, W_{[N]}\big) && \text{(81b)} \\ &= \sum_{k' \in [K]\setminus\{k\}} I\big(Q_{1,k'}; Q_{2,k'} | W_{[N]}\big) = 0 && \text{(81c)} \end{aligned}$$

where again (81a) follows from (4), (81b) follows from the fact that $\mathscr{M}_k = Q_{1,k}, k \in [K]$ and that $\{Y_{\mathcal{S}}, \mathcal{S} \subseteq [K], |\mathcal{S}| = t+1\}$ is a function of $\{Q_{2,k}, k \in [K]\}$ and $W_{[N]}$, and (81c) follows from the fact that $\mathscr{M}_k = Q_{1,k}$ is contained in Z_k and that the pairs $(Q_{1,k'}, Q_{2,k'}), k' \in [K]$ are independent of each other and that the query independence condition in (15) holds for the PIR scheme. This completes the proof.

TABLE III
PROPOSED PIR SCHEME FOR $N = 3$

	Server 1	Server 2		
		$d=1$	$d=2$	$d=3$
$T=0$	W_1+W_2	W_2	W_1	W_3
$T=1$	W_1+W_3	W_3	W_2	W_1
$T=2$	W_2+W_3	W_1	W_3	W_2

APPENDIX C
TWO-SERVER PIR SCHEMES FOR THEOREM 9

For the case $N = 2$, we use the proposed PIR scheme in [47, Section III-A]. Thus part 1 of the theorem is already proved. We proceed for other values.

A. $N = 3$

Assume the library has three files W_1, W_2, W_3. We define a random variable T which takes value uniformly at random from the set $\{0, 1, 2\}$. The proposed PIR scheme for different parameter regimes T and demand index d is depicted in Table III.

As one can see in Table III, there are 3 different answers for queries sent to Server 1 including $W_1 + W_2$, $W_1 + W_3$, and $W_2 + W_3$. We assign query values $Q_1 = 1$, $Q_1 = 2$, and $Q_1 = 3$ to these answers, respectively. Similarly, we assign query values $Q_2 = 1$, $Q_2 = 2$, and $Q_2 = 3$ for the answers of the second server W_1, W_2, and W_3 respectively. Note that the queries in the proposed scheme are independent of file realization, so we can remove the terms in the condition from the constraints in (13) and (15).

The query $Q_1 \in \{1, 2, 3\}$ sent to Server 1 is clearly independent of the demand. For the query $Q_2 \in \{1, 2, 3\}$ sent to Server 2 we have

$$\begin{aligned}\Pr(Q_2 = 1) &= \Pr(T = 0, d = 2) + \Pr(T = 1, d = 3) \\ &\quad + \Pr(T = 2, d = 1) \\ &= 1/3.\end{aligned}$$

In addition, we have

$$\Pr(Q_2 = 1|d = 1) = \Pr(T = 2|d = 1) = 1/3.$$

Hence, we will have $\Pr(Q_2 = 1) = \Pr(Q_2 = 1|d = 1)$. Following similarly, we can conclude that $P(Q_2) = P(Q_2|d)$ for all values $Q_2 \in \{1, 2, 3\}$ and $d \in \{1, 2, 3\}$, proving (13) to hold. Next, we should check the query independence condition in (15). We have

$$\begin{aligned}\Pr(Q_2 = 1|Q_1 = 1) &= \Pr(Q_2 = 1|T = 0) \\ &= 1/3 \\ &= \Pr(Q_2 = 1).\end{aligned}$$

Again, one can similarly show that $P(Q_2|Q_1) = P(Q_2)$ holds for all $Q_1 \in \{1, 2, 3\}$ and $Q_2 \in \{1, 2, 3\}$ proving (15) to hold. The decodability condition in (12) can also be easily checked to hold. The download cost from each server is 1, so **the achieved total download cost of this PIR scheme is $R_D = 2$, and the subpacketization is $F' = 1$.**

For the example of a coded caching with private demands and caches with parameters $N = 3, K = 2, M = 2$, using this PIR scheme in Theorem 5 with $N = 3$ and $t = 1$, we get the achieved load of $\frac{1}{2}$ and subpacketization level of 2. In this example, the achieved load by the virtual users scheme in [26] is $\frac{2}{5}$ and the needed subpacketization level is 15.

B. $N = 4$

Assume the library has four files W_1, W_2, W_3, W_4. We define a random variable T which takes a value uniformly at random from the set $\{0, 1, 2, 3\}$. The proposed PIR scheme for different parameter regimes T and demand index d is depicted in Table IV.

As one can see in Table IV, there are 4 different answers for queries sent to Server 1 including $W_1 + W_2 + W_3 + W_4$, $-W_1-W_2+W_3+W_4$, $-W_1+W_2-W_3+W_4$, and $-W_1+W_2+W_3-W_4$. We assign query values $Q_1 = 1$, $Q_1 = 2$, $Q_1 = 3$, and $Q_1 = 4$ to these answers respectively. Similarly we assign query values $Q_2 = 1$, $Q_2 = 2$, $Q_2 = 3$, and $Q_2 = 4$ for the answers of the second server $-W_1 + W_2 + W_3 + W_4$, $W_1 - W_2 + W_3 + W_4$, $W_1 + W_2 - W_3 + W_4$, and $W_1 + W_2 + W_3 - W_4$ respectively. The queries in the proposed scheme are independent of file realization, so we can remove the terms in the condition from the constraints in (13) and (15).

The query $Q_1 \in \{1, 2, 3, 4\}$ sent to Server 1 is clearly independent of the demand. For the query $Q_2 \in \{1, 2, 3, 4\}$ sent to Server 2 we have

$$\begin{aligned}\Pr(Q_2 = 1) &= \Pr(T = 0, d = 1) + \Pr(T = 1, d = 2) \\ &\quad + \Pr(T = 2, d = 3) + \Pr(T = 3, d = 4) \\ &= 1/4.\end{aligned}$$

In addition, we have

$$\Pr(Q_2 = 1|d = 1) = \Pr(T = 0|d = 1) = 1/4.$$

Hence, we will have $\Pr(Q_2 = 1) = \Pr(Q_2 = 1|d = 1)$. Following similarly, we can conclude that $P(Q_2) = P(Q_2|d)$ for all values $Q_2 \in \{1, 2, 3, 4\}$ and $d \in \{1, 2, 3, 4\}$, proving (13) to hold. Next we should check the query independence condition in (15). We have

$$\begin{aligned}\Pr(Q_2 = 1|Q_1 = 1) &= \Pr(Q_2 = 1|T = 0) \\ &= 1/4 \\ &= \Pr(Q_2 = 1).\end{aligned}$$

Again, one can similarly show that $P(Q_2|Q_1) = P(Q_2)$ holds for all $Q_1 \in \{1, 2, 3, 4\}$ and $Q_2 \in \{1, 2, 3, 4\}$ proving (15) to hold. The decodability condition in (12) can also be easily checked to hold. The download cost from each server is 1, so **the achieved total download cost of this PIR scheme is $R_D = 2$, and the subpacketization is $F' = 1$.**

For the example of a coded caching with private demands and caches with parameters $N = 4, K = 2, M = \frac{5}{2}$, using this PIR scheme in Theorem 5 with $N = 4$ and $t = 1$, we get the achieved load of $\frac{1}{2}$ and subpacketization level of 2. In this example, the achieved load by the virtual users scheme in [26] is $\frac{1}{2}$ and the needed subpacketization level is 56.

TABLE IV
PROPOSED PIR SCHEME FOR $N = 4$

	Server 1	Server 2			
		$d=1$	$d=2$	$d=3$	$d=4$
$T=0$	$W_1+W_2+W_3+W_4$	$-W_1+W_2+W_3+W_4$	$W_1-W_2+W_3+W_4$	$W_1+W_2-W_3+W_4$	$W_1+W_2+W_3-W_4$
$T=1$	$-W_1-W_2+W_3+W_4$	$W_1-W_2+W_3+W_4$	$-W_1+W_2+W_3+W_4$	$W_1+W_2+W_3-W_4$	$W_1+W_2-W_3+W_4$
$T=2$	$-W_1+W_2-W_3+W_4$	$W_1+W_2-W_3+W_4$	$W_1+W_2+W_3-W_4$	$-W_1+W_2+W_3+W_4$	$W_1-W_2+W_3+W_4$
$T=3$	$-W_1+W_2+W_3-W_4$	$W_1+W_2+W_3-W_4$	$W_1+W_2-W_3+W_4$	$W_1-W_2+W_3+W_4$	$-W_1+W_2+W_3+W_4$

APPENDIX D
PROOF OF THEOREM 8: LOWER BOUND ON TWO-SERVER PIR SCHEMES SATISFYING THE UDIQ CONDITION

A. Proof of Theorem 8-1

Without loss of generality, we assume that $\mathcal{Q}_1 = \{1, 2, \ldots, N_1\}$ and $\mathcal{Q}_2 = \{1, 2, \ldots, N_2\}$. Based on the fact that the queries should not reveal any information about the demand as in (13), we have

$$\Pr(d = \tau | Q_1 = 1) = \ldots = \Pr(d = \tau | Q_1 = N_1), \ \forall \tau \in [N], \tag{82}$$

where d is the demand. Based on the definition in (24), we further extend (82) as follows,

$$\sum_{Q_2 \in \mathcal{U}_{\tau|Q_1=1}} \Pr(Q_2|Q_1 = 1) = \ldots = \sum_{Q_2 \in \mathcal{U}_{\tau|Q_1=N_1}} \Pr(Q_2|Q_1 = N_1). \tag{83}$$

Because of the independent queries condition in (15), and because we have a uniform query distribution, the values of the probability functions $\Pr(Q_2|Q_1)$ for all Q_2 and Q_1 are the same, equal to $1/N_2$. So from (83) we have

$$\frac{|\mathcal{U}_{\tau|Q_1=1}|}{N_2} = \ldots = \frac{|\mathcal{U}_{\tau|Q_1=N_1}|}{N_2}. \tag{84}$$

This proves that $|\mathcal{U}_{\tau|Q_1=1}| = \cdots = |\mathcal{U}_{\tau|Q_1=N_1}|$. In addition to

$$\Pr(d = 1|Q_1 = 1) = \Pr(d = 2|Q_1 = 1) = \cdots = \Pr(d = N|Q_1 = 1),$$

which follows from the privacy constraint, we have $|\mathcal{U}_{\tau_1|Q_1=1}| = |\mathcal{U}_{\tau_2|Q_1=1}|$ where $\tau_1, \tau_2 \in [N]$. Similarly, we also have $|\mathcal{U}_{\tau|Q_2=1}| = \cdots = |\mathcal{U}_{\tau|Q_2=N_2}|$ and $|\mathcal{U}_{\tau_1|Q_2=1}| = |\mathcal{U}_{\tau_2|Q_2=1}|$ where $\tau_1, \tau_2 \in [N]$. This completes the proof of the first part.

B. Proof of Theorem 8-2

Based on the condition of independent queries, for any $q_2 \in \mathcal{Q}_2$, all the queries in $\mathcal{Q}_1$ should be exhausted for all choices of the demanded file index τ. In other words, for any $q_2 \in \mathcal{Q}_2$ we should have

$$N_1 \leq |\mathcal{U}_{\tau=1|Q_2=q_2}| + \cdots + |\mathcal{U}_{\tau=N|Q_2=q_2}| = Nn_1, \tag{85}$$

which resluts $\frac{N_1}{n_1} \leq N$. With the same argument, we have $\frac{N_2}{n_2} \leq N$.

C. Proof of Theorem 8-3

Based on the definition of $\mathcal{U}_{\tau|Q_1=q_1}$, we have

$$\mathcal{U}_\tau = \{(1, j_1) : j_1 \in \mathcal{U}_{\tau|Q_1=1}\} \bigcup \ldots \bigcup \{(N_1, j_{N_1}) : j_{N_1} \in \mathcal{U}_{\tau|Q_1=N_1}\} \tag{86}$$

Since all the sets above are disjoint and of size n_2, we have

$$|\mathcal{U}_\tau| = N_1 n_2, \tag{87}$$

or similarly

$$|\mathcal{U}_\tau| = N_2 n_1. \tag{88}$$

So we have

$$|\mathcal{U}_1| + |\mathcal{U}_2| + \ldots + |\mathcal{U}_N| = NN_1n_2 = NN_2n_1. \tag{89}$$

Since in total we have N_1N_2 different pairs of queries for the two servers, roughly speaking, each query pair should be able to decode $\frac{NN_1n_2}{N_1N_2} = N\frac{n_2}{N_2} = N\frac{n_1}{N_1} \triangleq \alpha N$ files. Thus, we would need at least $\frac{N}{\alpha N} = \frac{1}{\alpha} \triangleq \alpha'$ pairs of queries to cover all the files. A formal proof starts with the following lemma.

Lemma 1: For $\alpha_1 \in [N_1]$ and $\alpha_2 \in [N_2]$ such that $\alpha_1\alpha_2 = \left\lceil \frac{N_1}{n_1} \right\rceil = \left\lceil \frac{N_2}{n_2} \right\rceil$, there exist α_1 queries chosen from $\mathcal{Q}_1$ and α_2 queries chosen from $\mathcal{Q}_2$ such that the resulting $\alpha_1\alpha_2$ pairs of queries can recover all the N files.

Proof: We choose α_1 queries from $\mathcal{Q}_1$ and α_2 queries from $\mathcal{Q}_2$ uniformly at random. Without loss of generality, we assume that the chosen queries are $\mathcal{Q}_{1,\alpha_1} = \{1, 2, \ldots, \alpha_1\}$ and $\mathcal{Q}_{2,\alpha_2} = \{1, 2, \ldots, \alpha_2\}$ respectively. We should mention that because of the demand privacy constraint in (13), all the queries in $\mathcal{Q}_1$ and $\mathcal{Q}_2$ should appear at least once in $\mathcal{U}_\tau$ for any $\tau \in [N]$. For the first query from the first server $Q_1 = 1$, the probability that $Q_2 = 1$ is not in the set $\mathcal{U}_{\tau|Q_1=1}$ equals $\Pr(Q_2 = 1 \notin \mathcal{U}_{\tau|Q_1=1}) = 1 - \frac{n_2}{N_2}$. If we know that $Q_2 = 1 \notin \mathcal{U}_{\tau|Q_1=1}$, the probability that $Q_2 = 2 \notin \mathcal{U}_{\tau|Q_1=1}$ would be $\Pr(Q_2 = 2 \notin \mathcal{U}_{\tau|Q_1=1} | Q_2 = 1 \notin \mathcal{U}_{\tau|Q_1=1}) = 1 - \frac{n_2}{N_2-1}$. Similarly continuing, we can compute the probability that none of the α_2 queries chosen from $\mathcal{Q}_2$ appear as a pair with $Q_1 = 1$ in the set $\mathcal{U}_\tau$.

$$\begin{aligned}&\Pr\left(\{(Q_1 = 1, \mathcal{Q}_{2,\alpha_2})\} \cap \mathcal{U}_{\tau|Q_1=1} = \emptyset\right) \\ &= \left(1 - \frac{n_2}{N_2}\right)\left(1 - \frac{n_2}{N_2 - 1}\right) \ldots \left(1 - \frac{n_2}{N_2 - (\alpha_2 - 1)}\right) \\ &\leq \left(1 - \frac{n_2}{N_2}\right)^{\alpha_2}. \end{aligned} \tag{90}$$

Using the same argument for all α_1 queries from $\mathcal{Q}_1$, we have

$$\Pr\left((\mathcal{Q}_{1,\alpha_1}, \mathcal{Q}_{2,\alpha_2}) \cap \mathcal{U}_\tau = \emptyset\right) \tag{91a}$$

$$\leq \left(\left(1-\frac{n_2}{N_2}\right)^{\alpha_2}\right)^{\alpha_1} = \left(1-\frac{n_2}{N_2}\right)^{\alpha_1\alpha_2} \leq \left(1-\frac{n_2}{N_2}\right)^{\frac{N_2}{n_2}} \tag{91b}$$

$$\leq 1 - \frac{N_2}{n_2}\frac{n_2}{N_2} + o\left(\frac{n_2}{N_2}\right) \tag{91c}$$

$$= o\left(\frac{n_2}{N_2}\right), \tag{91d}$$

where (91c) follows from the Taylor expansion $(1-x)^y = 1 - yx + o(x)$. Now we can write the probability that the set $(\mathcal{Q}_{1,\alpha_1}, \mathcal{Q}_{2,\alpha_2})$ cannot decode at least one of the N files.

$$\Pr\left((\mathcal{Q}_{1,\alpha_1}, \mathcal{Q}_{2,\alpha_2}) \cap \mathcal{U}_\tau \neq \emptyset, \forall \tau \in [N]\right) \geq \left(1 - o\left(\frac{n_2}{N_2}\right)\right)^N > 0 \tag{92}$$

Since we have chosen our sets of queries randomly, and the probability that all the files are covered is greater than zero in a finite probability space, we can conclude that there exists at least one choice of α_1 queries from $\mathcal{Q}_1$ and one choice of α_2 queries from $\mathcal{Q}_2$ that covers all files. □

The proof of the third part of the theorem is follows directly from Lemma 1. As a result of Lemma 1, suppose we choose $\alpha_1 \in [N_1]$ queries from $\mathcal{Q}_1$ and $\alpha_2 \in [N_2]$ queries from $\mathcal{Q}_2$ such that the resulting number of pairs $\alpha_1\alpha_2 = \left\lceil \frac{N_1}{n_1} \right\rceil = \left\lceil \frac{N_2}{n_2} \right\rceil$ can recover all the files. Based on the cut-set bound we have

$$\alpha_1 R_{D_1} + \alpha_2 R_{D_2} \geq N. \tag{93}$$

Taking the minimum on the left hand-side, proves this part.

D. Proof of Theorem 8-4

For the forth part of the theorem, if we assume that $R_{D_1} = R_{D_2} = R'_D$ and $\alpha_1\alpha_2 = \lceil \alpha' \rceil$, we will have

$$R'_D \geq \frac{N}{\alpha_1 + \alpha_2} = \frac{N}{\alpha_1 + \frac{\lceil \alpha' \rceil}{\alpha_1}} = \frac{\alpha_1 N}{\alpha_1^2 + \lceil \alpha' \rceil}. \tag{94}$$

Thus we have

$$R'_D \geq \max_{\alpha_1 \in [N_1]} \frac{\alpha_1 N}{\alpha_1^2 + \lceil \alpha' \rceil}. \tag{95}$$

To derive the optimal value for α_1, we assume that it is continuous and take the derivative of the right hand side with respect to α_1 and put it equal to zero. We will have

$$N(\alpha_1^2 + \lceil \alpha' \rceil) = (\alpha_1 N)(2\alpha_1), \tag{96}$$

which results in $\alpha_1 = \sqrt{\lceil \alpha' \rceil}$. Since α_1 and α_2 should be integers, we can lower bound the right-hand side of (95) as follows,

$$R'_D \geq \frac{N}{2(\sqrt{\lceil \alpha' \rceil} + 1)} = \frac{N}{2\left(\sqrt{\left\lceil \frac{N_1}{n_1} \right\rceil} + 1\right)} = \frac{N}{2\left(\sqrt{\left\lceil \frac{N_2}{n_2} \right\rceil} + 1\right)}. \tag{97}$$

REFERENCES

[1] A. Gholami, K. Wan, H. Sun, M. Ji, and G. Caire, "Coded caching with private demands and caches," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2022, pp. 1396–1401.

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

[3] 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.

[4] 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.

[5] M. A. Maddah-Ali and U. Niesen, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Trans. Netw.*, vol. 23, no. 4, pp. 1029–1040, Aug. 2015.

[6] R. Pedarsani, M. A. Maddah-Ali, and U. Niesen, "Online coded caching," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 836–845, Apr. 2016.

[7] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless D2D networks," *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 849–869, Feb. 2016.

[8] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Order-optimal rate of caching and coded multicasting with random demands," *IEEE Trans. Inf. Theory*, vol. 63, no. 6, pp. 3923–3949, Jun. 2017.

[9] U. Niesen and M. A. Maddah-Ali, "Coded caching with nonuniform demands," *IEEE Trans. Inf. Theory*, vol. 63, no. 2, pp. 1146–1158, Feb. 2017.

[10] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. N. Diggavi, "Hierarchical coded caching," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 3212–3229, Jun. 2016.

[11] K. Shanmugam, M. Ji, A. M. Tulino, J. Llorca, and A. G. Dimakis, "Finite-length analysis of caching-aided coded multicasting," *IEEE Trans. Inf. Theory*, vol. 62, no. 10, pp. 5524–5537, Oct. 2016.

[12] S. Jin, Y. Cui, H. Liu, and G. Caire, "A new order-optimal decentralized coded caching scheme with good performance in the finite file size regime," *IEEE Trans. Commun.*, vol. 67, no. 8, pp. 5297–5310, Aug. 2019.

[13] Q. Yan, M. Cheng, X. Tang, and Q. Chen, "On the placement delivery array design for centralized coded caching scheme," *IEEE Trans. Inf. Theory*, vol. 63, no. 9, pp. 5821–5833, Sep. 2017.

[14] J. Wang, M. Cheng, Q. Yan, and X. Tang, "Placement delivery array design for coded caching scheme in D2D networks," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3388–3395, May 2019.

[15] Q. Yan, M. Wigger, and S. Yang, "Placement delivery array design for combination networks with edge caching," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018, pp. 1555–1559.

[16] S. Sasi and B. S. Rajan, "Multi-access coded caching scheme with linear sub-packetization using PDAs," *IEEE Trans. Commun.*, vol. 69, no. 12, pp. 7974–7985, Dec. 2021.

[17] M. Cheng, J. Wang, X. Zhong, and Q. Wang, "A framework of constructing placement delivery arrays for centralized coded caching," *IEEE Trans. Inf. Theory*, vol. 67, no. 11, pp. 7121–7131, Nov. 2021.

[18] X. Zhong, M. Cheng, and J. Jiang, "Placement delivery array based on concatenating construction," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1216–1220, Jun. 2020.

[19] C. Shangguan, Y. Zhang, and G. Ge, "Centralized coded caching schemes: A hypergraph theoretical approach," *IEEE Trans. Inf. Theory*, vol. 64, no. 8, pp. 5755–5766, Aug. 2018.

[20] K. Shanmugam, A. M. Tulino, and A. G. Dimakis, "Coded caching with linear subpacketization is possible using Ruzsa–Szeméredi graphs," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 1237–1241.

[21] Q. Yan, X. Tang, Q. Chen, and M. Cheng, "Placement delivery array design through strong edge coloring of bipartite graphs," *IEEE Commun. Lett.*, vol. 22, no. 2, pp. 236–239, Feb. 2018.

[22] L. Tang and A. Ramamoorthy, "Coded caching schemes with reduced subpacketization from linear block codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 4, pp. 3099–3120, Apr. 2018.

[23] M. Cheng, J. Jiang, X. Tang, and Q. Yan, "Some variant of known coded caching schemes with good performance," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1370–1377, Mar. 2020.

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

[25] F. Engelmann and P. Elia, "A content-delivery protocol, exploiting the privacy benefits of coded caching," in *Proc. 15th Int. Symp. Model. Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, May 2017, pp. 1–6.

[26] S. Kamath, "Demand private coded caching," 2019, *arXiv:1909.03324*.

[27] V. R. Aravind, P. K. Sarvepalli, and A. Thangaraj, "Lifting constructions of PDAs for coded caching with linear subpacketization," 2020, *arXiv:2007.07475*.

[28] Q. Yan and D. Tuninetti, "Fundamental limits of caching for demand privacy against colluding users," *IEEE J. Sel. Areas Inf. Theory*, vol. 2, no. 1, pp. 192–207, Mar. 2021.

[29] K. Wan, H. Sun, M. Ji, D. Tuninetti, and G. Caire, "On the optimal load-memory tradeoff of cache-aided scalar linear function retrieval," *IEEE Trans. Inf. Theory*, vol. 67, no. 6, pp. 4001–4018, Jun. 2021.

[30] S. Kamath, J. Ravi, and B. K. Dey, "Demand-private coded caching and the exact trade-off for N=K=2," in *Proc. Nat. Conf. Commun. (NCC)*, Feb. 2020, pp. 1–6.

[31] C. Gurjarpadhye, J. Ravi, S. Kamath, B. K. Dey, and N. Karamchandani, "Fundamental limits of demand-private coded caching," *IEEE Trans. Inf. Theory*, vol. 68, no. 6, pp. 4106–4134, Jun. 2022.

[32] V. R. Aravind, P. K. Sarvepalli, and A. Thangaraj, "Subpacketization in coded caching with demand privacy," in *Proc. Nat. Conf. Commun. (NCC)*, Feb. 2020, pp. 1–6.

[33] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," *J. ACM*, vol. 45, no. 6, pp. 965–981, 1998.

[34] G. Cohen, I. Honkala, S. Litsyn, and A. Lobstein, *Covering Codes*. Amsterdam, The Netherlands: Elsevier, 1997.

[35] A. Ambainis, "Upper bound on the communication complexity of private information retrieval," in *Proc. Int. Colloq. Automata, Lang., Program.* Berlin, Germany: Springer, 1997, pp. 401–407.

[36] T. Itoh, "Efficient private information retrieval," *IEICE Trans. Fundamentals Electron., Commun. Comput. Sci.*, vol. 82, no. 1, pp. 11–20, 1999.

[37] A. Beimel and Y. Ishai, "Information-theoretic private information retrieval: A unified construction," in *Proc. Int. Colloq. Automata, Lang., Program.* Berlin, Germany: Springer, 2001, pp. 912–926.

[38] A. A. Razborov and S. Yekhanin, "An $\Omega(n^{1/3})$ lower bound for bilinear group based private information retrieval," in *Proc. 47th Annu. IEEE Symp. Found. Comput. Sci. (FOCS)*, Oct. 2006, pp. 739–748.

[39] O. Goldreich, H. Karloff, L. J. Schulman, and L. Trevisan, "Lower bounds for linear locally decodable codes and private information retrieval," in *Proc. 17th IEEE Annu. Conf. Comput. Complex.*, 2005, pp. 1424–1436.

[40] A. Chakrabarti and A. Shubina, "Nearly private information retrieval," in *Proc. Int. Symp. Math. Found. Comput. Sci.* Berlin, Germany: Springer, 2007, pp. 383–393.

[41] R. Beigel, L. Fortnow, and W. Gasarch, "A nearly tight lower bound for private information retrieval protocols," *Electron. Colloquim Comput. Complex. (ECCC)*, 2003.

[42] D. Woodruff and S. Yekhanin, "A geometric approach to information-theoretic private information retrieval," in *Proc. 20th Annu. IEEE Conf. Comput. Complex. (CCC)*, 2005, pp. 275–284.

[43] Z. Dvir and S. Gopi, "2-server PIR with subpolynomial communication," *J. ACM*, vol. 63, no. 4, pp. 1–15, Nov. 2016.

[44] K. Efremenko, "3-query locally decodable codes of subexponential length," *SIAM J. Comput.*, vol. 41, no. 6, pp. 1694–1703, Jan. 2012.

[45] S. Yekhanin, "Towards 3-query locally decodable codes of subexponential length," *J. ACM*, vol. 55, no. 1, pp. 1–16, Feb. 2008.

[46] 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.

[47] C. Tian, H. Sun, and J. Chen, "Capacity-achieving private information retrieval codes with optimal message size and upload cost," *IEEE Trans. Inf. Theory*, vol. 65, no. 11, pp. 7613–7627, Nov. 2019.

[48] K. Banawan and S. Ulukus, "Multi-message private information retrieval: Capacity results and near-optimal schemes," *IEEE Trans. Inf. Theory*, vol. 64, no. 10, pp. 6842–6862, Oct. 2018.

[49] H. Sun and S. A. Jafar, "The capacity of symmetric private information retrieval," *IEEE Trans. Inf. Theory*, vol. 65, no. 1, pp. 322–329, Jan. 2019.

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

[51] R. Tajeddine, O. W. Gnilke, and S. E. Rouayheb, "Private information retrieval from MDS coded data in distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 64, no. 11, pp. 7081–7093, Nov. 2018.

[52] 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.

[53] S. P. Shariatpanahi, M. J. Siavoshani, and M. A. Maddah-Ali, "Multi-message private information retrieval with private side information," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Nov. 2018, pp. 1–5.

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

[55] 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.

[56] N. B. Shah, K. V. Rashmi, and K. Ramchandran, "One extra bit of download ensures perfectly private information retrieval," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2014, pp. 856–860.

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

[58] I. Samy, R. Tandon, and L. Lazos, "On the capacity of leaky private information retrieval," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 1262–1266.

[59] C. Dwork, "Differential privacy: A survey of results," in *Proc. Int. Conf. Theory Appl. Models Comput.* Berlin, Germany:Springer, 2008, pp. 1–19.

[60] H.-Y. Lin, S. Kumar, E. Rosnes, A. G. I. Amat, and E. Yaakobi, "Weakly-private information retrieval," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 1257–1261.

[61] H.-Y. Lin, S. Kumar, E. Rosnes, A. G. I. Amat, and E. Yaakobi, "The capacity of single-server weakly-private information retrieval," *IEEE J. Sel. Areas Inf. Theory*, vol. 2, no. 1, pp. 415–427, Mar. 2021.

[62] I. Samy, M. Attia, R. Tandon, and L. Lazos, "Asymmetric leaky private information retrieval," *IEEE Trans. Inf. Theory*, vol. 67, no. 8, pp. 5352–5369, Aug. 2021.

[63] T. Guo, R. Zhou, and C. Tian, "On the information leakage in private information retrieval systems," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2999–3012, 2020.

[64] R. Zhou, T. Guo, and C. Tian, "Weakly private information retrieval under the maximal leakage metric," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2020, pp. 1089–1094.

[65] A. El Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[66] V. Grolmusz, "Superpolynomial size set-systems with restricted intersections mod 6 and explicit Ramsey graphs," *Combinatorica*, vol. 20, no. 1, pp. 71–86, Jan. 2000.

Ali Gholami (Student Member, IEEE) received the B.Sc. degree in electrical engineering from the University of Tehran, Iran, in 2016, and the M.Sc. degree in electrical engineering from the Sharif University of Technology, Iran, in 2018. He is currently pursuing the Ph.D. degree with the Communications and Information Theory (CommIT) Group, Technical University of Berlin, Germany. His research interests include information theory, coding theory, and privacy.

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. From 2018 to 2022, he was a Post-Doctoral Researcher with the Communications and Information Theory Chair (CommIT), Technische Universität Berlin, Berlin, Germany. He is currently a Professor with the School of Electronic Information and Communications, Huazhong University of Science and Technology. His research interests include information theory, coding techniques, and their applications on coded caching, index coding, distributed storage, distributed computing, wireless communications, privacy, and security. He received the Best Young Scientist Award from the 8th International Conference on Computer and Communication Systems in 2023. He has been serving as an Associate Editor for IEEE COMMUNICATIONS LETTERS since August 2021.

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. and Ph.D. degrees in electrical and computer engineering from the University of California at Irvine, Irvine, CA, USA, in 2013 and 2017, respectively.

He is currently an Associate Professor with the Department of Electrical Engineering, University of North Texas, Denton, TX, 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, the UNT College of Engineering Junior Faculty Research Award in 2021, and the UNT College of Engineering Distinguished Faculty Fellowship in 2023. His coauthored papers received the IEEE Jack Keil Wolf ISIT Student Paper Award in 2016, the IEEE GLOBECOM Best Paper Award in 2016, and the 2020–2021 IEEE Data Storage Best Student Paper Award.

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. degrees in electrical engineering from the KTH Royal Institute of Technology, Sweden, and the University of California, Santa Cruz, in 2008 and 2010, respectively, and the Ph.D. degree from the Ming Hsieh Department of Electrical Engineering, University of Southern California, in 2015. Subsequently, he was a Staff II System Design Scientist with Broadcom Corporation (Broadcom Ltd.) from 2015 to 2016. He is currently an Associate Professor with the Department of Electrical and Computer Engineering and an Adjunct Associate Professor with the School of Computing, The University of Utah. His research interests include information theory, coding theory, concentration of measure and statistics with the applications of distributed computing systems, wireless communications and networking, caching networks, distributed machine learning, distributed storage, and (statistical) signal processing. He received the NSF CAREER Award in 2022, the IEEE Communications Society Leonard G. Abraham Prize for the Best IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS Paper in 2019, the Best Paper Awards from the 2021 IEEE GLOBECOM Conference and the 2015 IEEE ICC Conference, the Best Student Paper Award from the 2010 IEEE European Wireless Conference, and the USC Annenberg Fellowship from 2010 to 2014. He has been serving as Associate Editors for IEEE TRANSACTIONS ON INFORMATION THEORY since 2022 and IEEE TRANSACTIONS ON COMMUNICATIONS since 2020.

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

He was a Post-Doctoral Research Fellow with the European Space Agency (ESTEC, Noordwijk, The Netherlands), from 1994 to 1995; an Assistant Professor of telecommunications with 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, Los Angeles, CA, USA. 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 communications theory, information theory, and channel and source coding with a particular focus on wireless communications. 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 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 JOURNAL ON SELECTED AREAS IN COMMUNICATIONS Paper in 2019, the IEEE Communications Society Edwin Howard Armstrong Achievement Award in 2020, and the 2021 Leibinz Prize of the German National Science Foundation (DFG). He has served on the Board of Governors for the IEEE Information Theory Society, from 2004 to 2007, and an Officer, from 2008 to 2013. He was the President of the IEEE Information Theory Society in 2011.