On Coded Caching with Private Demands

Kai Wan, *Member, IEEE,* and Giuseppe Caire, *Fellow, IEEE*

Abstract

Caching is an efficient way to reduce network traffic congestion during peak hours by storing some content at the user's local cache memory without knowledge of later demands. For the shared-link caching model, Maddah-Ali and Niesen (MAN) proposed a two-phase (*placement* and *delivery*) coded caching strategy, which is order optimal within a constant factor. However, in the MAN coded caching scheme, each user can obtain the information about the demands of other users, i.e., the MAN coded caching scheme is inherently prone to tampering and spying the activity/demands of other users. In this paper, we formulate an information-theoretic shared-link caching model with private demands, where there are K cache-aided users (which can cache up to M files) connected to a central server with access to N files. Each user requests L files. Our objective is to design a two-phase private caching scheme with minimum load while preserving the information-theoretic privacy of the demands of each user with respect to other users.

A trivial solution is the uncoded caching scheme which lets each user recover all the N files, referred to as *baseline scheme*. For this problem we propose two novel schemes which achieve the information-theoretic privacy of the users' demands while also achieving a non-trivial caching gain over the baseline scheme. The general underlying idea is to satisfy the users' requests by generating a set of coded multicast messages that is symmetric with respect to the library files, such that for each user k, the mutual information between these messages and the demands of all other users given the cache content and the demand of user k is zero. In the first scheme, referred to as virtual-user scheme, we introduce a number of virtual users such that each L-subset of files is demanded by K real or virtual (effective) users and use the MAN delivery to generate multicast messages. From the viewpoint of each user, the set of multicast messages is symmetric over all files even if each single multicast message is not. This scheme incurs in an extremely large sub-packetization. Then, we propose a second scheme, referred to as MDS-based scheme, based on a novel MDS-coded cache placement. In this case, we generate multicast messages where each multicast message contains one MDS-coded symbol from each file in the library and thus is again symmetric over all the files from the viewpoint of each user. The

sub-packetization level of the MDS-based scheme is exponentially smaller than that needed by the virtual-user scheme.

Compared with the existing shared-link coded caching converse bounds without privacy, the virtual-user scheme is proved to be order optimal with a constant factor when $N \leq LK$, or when $N \geq LK$ and $M \geq N/K$. In addition, when $M \geq N/2$, both of the virtual-user scheme and the MDS-based scheme are order optimal within a factor of 2.

Index Terms

Coded caching, information-theoretic privacy, virtual users, MDS code.

I. INTRODUCTION

A. Brief Review of Coded Caching

Recent years have witnessed a steep increase of wireless devices connected to the Internet, leading to a heavy network traffic because of multimedia streaming, web-browsing and social networking. Furthermore, the high temporal variability of network traffic results in congestions during peak-traffic times and underutilization of the network during off-peak times. *Caching* is a promising technique to reduce peak traffic by taking advantage of memories distributed across the network to duplicate content during off-peak times [1] . With the help of caching, network traffic could be shifted from peak to off-peak hours in order to smooth out the traffic load and reduce congestion. In the seminal paper [2], an information-theoretic and network-coding theoretic model for caching was proposed. In this model, two phases are included in a caching system: i) *placement phase*: each user equipped with cache stores some bits in its cache component without knowledge of later demands; ii) *delivery phase*: after each user has made its request and according to cache contents, the server transmits packets such that each user can recover its desired file(s). The goal is to minimize the number of transmitted bits (referred to as *load* in this paper).

Coded caching strategy was originally proposed in [2] for the shared-link broadcast networks where a server with a library of N files, of B bits each, is connected to K users (each of which is with a cache of MB bits) through a shared error-free broadcast link. Each user requests one file independently in the delivery phase. Maddah-Ali and Niesen (MAN) proposed a coded caching scheme that utilizes an uncoded combinatorial cache construction in the placement phase and a binary linear network code to generate multicast messages in the delivery phase. For $M = t\frac{N}{K}$ with

$t \in [0 : K]$, the transmitted load is $\frac{K(1-M/N)}{1+KM/N}$. For other memory size, the lower convex envelope of the above memory-load tradeoff points is achievable by memory-sharing between schemes for integer values of $t = KM/N$. Compared to the conventional uncoded caching scheme which lets each user store MB/N bits of each file in the placement phase and broadcasts the uncached part of each desired file during the delivery phase with the transmitted load $K(1 - M/N)$, the MAN coded caching scheme has an additional coded caching gain (i.e., load reduction factor) equal to $1 + KM/N$. It was proved in [3] that the worst-case load achieved by the MAN coded caching scheme among all possible demands is optimal under the constraint of uncoded placement (i.e., each user directly stores a subset of bits in the library) and $N \geq K$. When $N \geq K$, the MAN coded caching scheme was also proved in [4] to be generally order optimal within a factor of 2. For any N and K, a factor of 4 for the order optimality of the MAN coded caching scheme was proved in [5]. By observing that some MAN multicast messages are redundant for the case $N < K$, the authors in [6] proposed an improved delivery scheme which is optimal under the constraint of uncoded cache placement for any N and K, and optimal within a factor of 2 over all possible placement strategies.

In the MAN caching model, each user requests only one file, which may not be practical. The caching problem with multi-request was originally considered in [7] where each user demands L files from the library. With the MAN placement, to divide the delivery phase into L rounds where in each round the MAN coded caching scheme in [2] (referred to as L-*round MAN coded caching scheme* in this paper, which reduces to the MAN coded caching scheme when $L = 1$) is used to let each user decode one file, can achieve a generally order optimal worst-case load within a factor of 18 [7]. By further tightening the converse bound, this order optimality factor was reduced to 11 in [8].

The MAN coded caching strategy was also used in a number of extended models, such as decentralized caching where users must fill their caches independently of other users [9], device-to-device (D2D) caching systems where users communicate among each other during the delivery phase [10], cache-aided topological networks where the server communicates with the users through some intermediate relays [11]–[13], etc. However, these extended models will not be considered in our paper, and thus we do not go into details.

B. Existing Secure Coded Caching Schemes

Soon after the appearance of [2], various 'secure' versions of the caching problem have been proposed. Secure coded caching was originally considered in [14], where there are some wiretappers who can also receive the broadcasted packets from the server. To prevent the wiretappers from obtaining any information about the files in the library, the authors in [14] let each user store not only the content about the library in its cache, but also some 'keys'. In the delivery phase, each multicast message is generated by taking XOR of the MAN multicast message and some key in order to 'lock' the multicast messages such that only the intended users can unlock it. This secure caching scheme against wiretappers was proved in [15] to be optimal under the constraint of uncoded cache placement. Another secure shared-link caching model was proposed in [16]. In this case, the objective is to avoid each user to get any information about the files not required by that user. The placement and delivery phases were designed based on the MAN coded caching scheme with an additional secret sharing precoding [17] on each file (i.e., by encoding a message with (n, t) secret sharing code where $n > t$, any t shares do not reveal any information about the message and the message can be reconstructed from all the n shares). In addition, the secure caching scheme in [16] could also successfully prevent external wiretappers, because each multicast message is also locked by a key. The above strategies to prevent external wiretappers and internal malicious users from retrieving information about the library, were then used in extended models, such as D2D caching systems [18], [19], topological cache-aided relay networks [20], erasure broadcast channels [21], etc.

C. Coded Caching with Private Demands

The existing secure caching schemes are based on the MAN coded caching scheme (with or without a secure precoding on each file) and then generate locked MAN multicast messages. However, a malicious user could simply use the MAN multicast messages (or locked MAN multicast messages) in order to learn the requests of other users, e.g., to perform some survey on user preferences, which is not good in terms of privacy. Shared-link caching problem with single request to preserve the users demands from other users was originally considered in [22]. The caching scheme proposed in [22] generates ℓ virtual users each of which randomly demands one file, such that each user cannot match the exact request to any other user. However, this caching scheme is not completely private from an information-theoretic viewpoint. For example, if there exists undemanded file by any real or virtual user, each user will know this file has

not been demanded such that it can get some information about the users demands from the transmission. In this paper, we formulate an information-theoretic caching problem which aims to preserve the privacy of the demands of each user with respect to other users during the transmission.

Let us focus on a toy example with $K = 2$, $N = 3$ and $M = 2N/3 = 2$. In this example, $t = KM/N = 4/3$, which is not an integer, and thus we should use the memory-sharing between $M_1 = Nt_1/K = 3/2$ with $t_1 = 1$ and $M_2 = Nt_2/K = 3$ with $t_2 = 2$. By the MAN placement, we divide each file into three equal-length and non-overlapping subfiles, the ith file, denoted by F_i, has three subfiles $F_{i,\{1\}}$, $F_{i,\{2\}}$, and $F_{i,\{1,2\}}$. User 1 caches $F_{i,\{1\}}$ and $F_{i,\{1,2\}}$, while user 2 caches $F_{i,\{2\}}$ and $F_{i,\{1,2\}}$.

In the delivery phase, we consider two demands:

- if the demand is $(1, 2)$, i.e., user 1 demands file F_1 and user 2 demands file F_2, we transmit the MAN multicast message $F_{1,\{2\}} \oplus F_{2,\{1\}}$, where $\oplus$ represents the XOR operation, such that user 1 can recover $F_{1,\{2\}}$ and user 2 can recover $F_{2,\{1\}}$. However, for the sake of successful decoding, user 1 needs to know that $F_{2,\{1\}}$ is contained by the multicast message, and thus it knows user 2 demands F_2. Similarly, user 2 will know the demand of user 1.

- if the demand is $(1, 1)$, i.e., both users 1 and 2 demand file F_1, we transmit the MAN multicast message $F_{1,\{2\}} \oplus F_{1,\{1\}}$, such that user 1 can recover $F_{1,\{2\}}$ and user 2 can recover $F_{1,\{1\}}$. However, from the transmission, user 1 knows user 2 demands F_1, while user 2 knows the demand of user 1.

The above example shows that the MAN scheme is inherently prone to tampering and spying the activity/demands of other users. In this paper we develop schemes that are able to provide full information-theoretic privacy of the users' demands, while still providing a non-trivial caching gain. To motivate the reader and show that this is indeed possible, we continue our toy example with the following scheme, which is a special case of the MDS-based scheme in Theorem 4. In the placement phase, we encode each file F_i by a $(4, 3)$ MDS code (i.e., each file F_i is split into 3 blocks of $B/3$ bits each, which are then encoded by a $(4, 3)$ MDS code such that each of the four MDS coded symbols has $B/3$ bits). Each file can be reconstructed by any three MDS coded symbols. The four MDS coded symbols are denoted by $S_1^i, S_2^i, S_3^i, S_4^i$. We randomly generate a permutation of $\{1, 2, 3, 4\}$, denoted by $\mathbf{p}_i = (p_{i,1}, p_{i,2}, p_{i,3}, p_{i,4})$ and let $F_{i,\emptyset} = S_{p_{i,1}}^i$, $F_{i,\{1\}} = S_{p_{i,2}}^i$, $F_{i,\{2\}} = S_{p_{i,3}}^i$, and $F_{i,\{1,2\}} = S_{p_{i,4}}^i$. We let user 1 cache $F_{i,\{1\}}$ and $F_{i,\{1,2\}}$, and user 2 cache $F_{i,\{2\}}$ and $F_{i,\{1,2\}}$. Notice that, for the sake of successful decoding, user 1 knows

the compositions of $F_{i,\{1\}}$ and $F_{i,\{1,2\}}$ (i.e., it knows from which code on which bits $F_{i,\{1\}}$ and $F_{i,\{1,2\}}$ are generated), but it does not know $\mathbf{p}_i$, i.e., it does not know which one of $F_{i,\{1\}}$ and $F_{i,\{1,2\}}$ is cached by user 2. Hence, $F_{i,\{1\}}$ and $F_{i,\{1,2\}}$ are equivalent from the viewpoint of user 1. Similarly, $F_{i,\{2\}}$ and $F_{i,\emptyset}$ are also equivalent from the viewpoint of user 1.

In the delivery phase, we also consider two demands:

- if the demand is $(1, 2)$, we transmit $F_{1,\{2\}} \oplus F_{2,\{1\}} \oplus F_{3,\{1,2\}}$, such that user 1 can recover $F_{1,\{2\}}$ and user 2 can recover $F_{2,\{1\}}$. Notice that each user only knows the composition of each MDS coded symbol in the sum, without knowing whether the other user caches it or not. From the viewpoint of user 1, in the sum there is one MDS coded symbol from each file and among these MDS coded symbols it caches the ones from the non-demanded files (i.e., F_2 and F_3).

- if the demand is $(1, 1)$, we transmit $F_{1,\emptyset} \oplus F_{2,\{1,2\}} \oplus F_{3,\{1,2\}}$, such that users 1 and 2 can recover $F_{1,\emptyset}$. Again, from the viewpoint of user 1, in the sum there is one MDS coded symbol from each file and among these MDS coded symbols, it caches the ones from the non-demanded files.

For the above two demands, from the viewpoint of user 1, the delivery phases are equivalent. Hence, user 1 cannot know any information about the request of user 2. A symmetric situation holds for user 2 and for all other possible demands.

In practice, it may be important to preserve the privacy of the users demands. The above example motivates the following question: what is the fundamental coded caching gain subject to such strict privacy constraint on the users demands? In this paper, we focus on the private shared-link caching model with multiple requests from an information-theoretic viewpoint, where each user requests L files. The objective is to design a private caching scheme with minimum load in the delivery phase, in order to maintain the successful decoding for each user and also to prevent each user from getting any information about other users' demands.

D. Relation to Private Information Retrieval

The privacy of the users demands was originally considered as the Private Information Retrieval (PIR) problem in [23]. In this setting, a user wants to retrieve a desired message from some distributed non-colluding databases (servers), and the objective is to prevent any server from retrieving any information about the users' demand. Recently, the authors in [24] characterized

the information-theoretic capacity of the PIR problem by proposing a novel converse bound and a coded PIR scheme based on an interference alignment idea.

Later, models combining the PIR problem with some caching component were proposed in [25]–[30]. In [25], the user randomly caches some files in the library and its side information is unknown to the servers. The capacity region of the rate in terms of the number of cached files was characterized in [25]. The authors extended the model in [25] to the single-server multi-user case, where each user caches some files and knows the demands of other users. A caching scheme based on Maximum Distance Separable (MDS) code was proposed. In [27]–[29], for the single-user PIR problem with end-user-cache, instead of caching the whole files, the user can choose any bits to store as in the coded caching model. Novel converse and achievable bounds were proposed in [27]–[29] for the cases where the user's cache is known, partially known, and unknown to the servers, respectively. The authors in [30] considered the single-user PIR problem with end-database-caches, where each server can choose any bits to store instead of being able to access to the whole library. Under the constraint of uncoded cache placement, the optimal PIR scheme was given in [30]. The PIR problem was then generalized to the Private Computation (PC) problem in [31], where the user should compute a function on the library instead of directly retrieving one message.

The considered coded caching problem with private demands aims to preserve the privacy of the demands of each user from other users, while the cache-aided PIR (or PC) problems aim to preserve the privacy of the users demands from the databases. Hence, the main challenge of the considered problem is to design multicast messages transmitted from the server such that each user can decode its desired files without getting any information of other users' demands, while still achieving a non-trivial coded caching gain.

E. Contributions

Our main contributions are as follows.

- **Problem formulation.** We formulate an information-theoretic shared-link coded caching model with multiple requests, and the constraints on the information-theoretic privacy of the users demands from other users.
- **Private coded caching schemes.** With a novel idea of private placement precoding (which makes the cached (resp. uncached) bits from each file equivalent from the viewpoint of each user), we then propose two private coded caching schemes with two different strategies to

generate a set of coded multicast messages which is symmetric over all the files (i.e., independent of the users' demands) from the viewpoint of each user.

1) Inspired by the virtual-user strategy originally introduced in [22], we propose a novel private caching scheme, referred to as *virtual user* scheme, by generating $\binom{N}{L}K - K$ virtual users such that each L-subset of files is demanded by exactly K real or virtual (effective) users. We then propose a private delivery scheme based on the $\binom{N}{L}K$-user MAN delivery scheme. Thus by 'hiding' the real users among all effective users, the set of coded multicast messages is symmetric over all the files and independent of the users' demands, from the viewpoint of each user. Notice that the caching scheme in [22] generates an arbitrary number of virtual users each of whom randomly demands one file, which cannot guarantee the information-theoretic privacy constraint even if the number of virtual users goes to infinity.

2) The main limitation of the virtual-user scheme is its sub-packetization level, which is equal to the sub-packetization level of the $\binom{N}{L}K$-user MAN coded caching scheme (it has the order $\mathcal{O}\left(2^{\binom{N}{L}K}\right)$ when $M \approx N/2$). In order to reduce the sub-packetization level, we propose the second scheme, referred to as *MDS-based* scheme. With a novel MDS-based cache placement, the main strategy is to generate multicast messages in the delivery phase, such that each multicast message contains one MDS-coded symbol from each file and thus is symmetric over all the N files from the viewpoint of each user. There is no MDS-coded symbol appearing in two multicast messages, which makes the set of all multicast messages also symmetric over all the N files. The needed sub-packetization level is $\mathcal{O}\left(2^{K}\right)$, which reduces exponentially the one of the virtual-user scheme and is the same as the maximal sub-packetization level of the K-user MAN coded caching scheme.

- **Order optimality results.** We summarize the order optimality results of the two proposed schemes in Table I. In short, the virtual-user scheme is order optimal within a constant factor when $N \leq LK$, or when $N \geq LK$ and $M \geq N/K$. In addition, when $M \geq N/2$, the virtual-user scheme and the MDS-based scheme have the same order optimality results.

F. Paper Organization

The rest of this paper is organized as follows. Section II formulates the considered shared-link caching model with private demands. Section III lists all the results in this paper and provide

Table I: Order optimality factors of the virtual-user scheme and the MDS-based scheme.

| | N > LK, M < N/2 | | N ≤ LK, M < N/2 | | M ≥ N/2 |
	L = 1	L > 1	L = 1	L > 1	
Virtual-user scheme	4, for $\frac{N}{K} \leq M < \frac{N}{2}$	22, for $\frac{N}{K} \leq M < \frac{N}{2}$	8	22	2
MDS-based scheme					2

some numerical evaluations. Section IV presents the proposed private caching schemes. Section V concludes the paper and some proofs are given in the Appendices.

G. Notation Convention

Calligraphic symbols denote sets, bold symbols denote vectors, and sans-serif symbols denote system parameters. We use $|\cdot|$ to represent the cardinality of a set or the length of a vector; $[a:b] := \{a, a+1, \ldots, b\}$ and $[n] := [1, 2, \ldots, n]$; $\oplus$ represents bit-wise XOR; $\mathbb{E}[\cdot]$ represents the expectation value of a random variable; $[a]^+ := \max\{a, 0\}$; we let $\binom{x}{y} = 0$ if $x < 0$ or $y < 0$ or $x < y$; we denote the power set of $[a]$ by $\text{Pow}(a)$, and sort all sets in lexicographic order. $\text{Pow}(a, j)$ denotes the jth set. For example,

$$\text{Pow}(3) = \{\emptyset, \{1\}, \{1, 2\}, \{1, 2, 3\}, \{1, 3\}, \{2\}, \{2, 3\}, \{3\}\},$$

and $\text{Pow}(3, 1) = \emptyset$, $\text{Pow}(3, 2) = \{1\}$, etc.

II. SYSTEM MODEL AND RELATED RESULTS

A. System Model

A (K, N, M, L) shared-link caching system with private demands is defined as follows. The system contains a server with access to a library of N independent files, denoted by $(F_1, F_2, \ldots, F_N)$, where each file is composed of B i.i.d. bits. As in [2], we assume that B is sufficiently large such that any sub-packetization of the files is possible. The server is connected to K users through an error-free shared-link. The caching system operates in two phases.

Placement Phase. During the placement phase, user $k \in [K]$ stores content in its cache of size MB bits without knowledge of later demands, where $M \in [0, N]$. We denote the content in the cache of user $k \in [K]$ by Z_k, which contains two parts

$$Z_k = (\mathcal{M}(C_k), C_k), \tag{1}$$

where C_k represents cached content from the N files,

$$C_k = \phi_k(F_1, \ldots, F_N, \mathscr{M}(C_k)), \tag{2}$$

and $\mathscr{M}(C_k)$ represents the metadata/composition of C_k (i.e., from which code on which bits, C_k are generated). For any bit in C_k, the metadata of this bit does not reveal which of the other users cache it. Notice that $\mathscr{M}(C_1), \ldots, \mathscr{M}(C_\mathsf{K})$ are random variables over $\mathcal{C}_1, \ldots, \mathcal{C}_\mathsf{K}$, representing all types of cache placement which can be used by the K users. In addition, for any $k \in [\mathsf{K}]$, the realization of $\mathscr{M}(C_k)$ is known by user k and is not known by other users.

We assume that the total length of $\mathscr{M}(C_k)$ compared to the file length B such that, for simplicity, the relevant cache size constraint is

$$\frac{H(Z_k)}{\mathsf{B}} = \frac{H(C_k)}{\mathsf{B}} \leq \mathsf{M}, \ \forall k \in [\mathsf{K}]. \text{ (Memory size)} \tag{3}$$

We also denote by $\mathbf{Z} := (Z_1, \ldots, Z_K)$ the content of all K caches.

Delivery Phase. During the delivery phase, each user demands L files, where $\mathsf{L} \in [\mathsf{N}]$. In this paper, we consider $\mathsf{N} \geq \mathsf{L}$ to ensure each user has L demands. The demand vector of user $k \in [\mathsf{K}]$ are denoted by $\mathbf{d}_k := (d_{k,1}, d_{k,2}, \ldots, d_{k,\mathsf{L}})$, where $1 \leq d_{k,1} < d_{k,2} < \cdots < d_{k,\mathsf{L}} \leq \mathsf{N}$. The demand matrix of all K users is denoted by $\mathbb{D} := [\mathbf{d}_1; \mathbf{d}_2; \ldots; \mathbf{d}_\mathsf{K}]$. In addition, we define $\mathbb{D}_{\backslash\{k\}}$ for each $k \in [\mathsf{K}]$ as the demand vectors of all users except user k, where

$$\mathbb{D}_{\backslash\{k\}} := [\mathbf{d}_1; \ldots; \mathbf{d}_{k-1}, \mathbf{d}_{k+1}, \ldots, \mathbf{d}_\mathsf{K}]. \tag{4}$$

We also denote the set of all possible demand matrices by

$$\mathscr{D} := \{\mathbb{D} : 1 \leq d_{k,1} < d_{k,2} < \cdots < d_{k,\mathsf{L}} \leq \mathsf{N}, \forall k \in [\mathsf{K}]\}. \tag{5}$$

We assume that the metadatas of users' caches, users' demands, and the library contents are independent,

$$H\big(F_1, F_2, \ldots, F_\mathsf{N}, \{\mathscr{M}(C_k) : k \in [\mathsf{K}]\}, \{T_\mathcal{S} : \mathcal{S} \subseteq [\mathsf{K}]\}, \mathbb{D}\big)$$
$$= \mathsf{NB} + H(\{\mathscr{M}(C_k) : k \in [\mathsf{K}]\}) + \sum_{\mathcal{S} \subseteq [\mathsf{K}]} H(T_\mathcal{S}) + H(\mathbb{D}). \tag{6}$$

Given the demand matrix $\mathbb{D}$ and the users' caches $\mathbf{Z}$, the server broadcasts a packet $X = (\mathscr{M}(P), P)$ which includes three parts (Header, Metadata, and Payload) as illustrated in Fig. 1. The header of X provides information (e.g., protocols, source, destination, etc.) to ensure that all users in $[\mathsf{K}]$ can receive successfully the broadcasted packet X. To ensure the successful decoding on the payload, the metadata $\mathscr{M}(P)$ represents the composition of the payload P.

Figure 1: The delivery packet of $X_{\mathcal{S}}$, where 'H' represents *Header*, 'M' represents *Metadata*, 'P' represents *Payload*.

Notice that $\mathscr{M}(P)$ is random variable over $\mathcal{P}$, representing all types of transmissions by the server. The payload contains the coded packets from the N files,

$$P = \psi\big(F_1, \ldots, F_\mathsf{N}, \mathscr{M}(P)\big). \tag{7}$$

We also assume that the total length of the header and metadata are negligible compared to the payload, such that we have

$$\mathsf{R} := H(X)/\mathsf{B} = H(P)/\mathsf{B}, \tag{8}$$

where R represents the load (i.e., normalized number of total transmitted bits) of X.

The constraints on the decoding of the demanded file by each user while maintaining the privacy is given as follows. For each user $k \in [\mathsf{K}]$, it must hold that

$$H(\{F_i : i \in \mathbf{d}_k\}|X, Z_k, \mathbf{d}_k) = 0, \ \forall k \in [\mathsf{K}]. \text{ (Decodability)} \tag{9}$$

In addition, given $\mathbf{d}_k$, user k cannot get any information about the demands of other users from X, i.e., the information-theoretic privacy constraint is

$$I(\mathbb{D}_{\backslash\{k\}}; X, Z_k|\mathbf{d}_k) = 0, \ \forall k \in [\mathsf{K}]. \text{ (Privacy)} \tag{10}$$

In other words, the mutual information between $\mathbb{D}_{\backslash\{k\}}$ and the user information after the delivery phase, quantifies in precise information-theoretic terms the information leakage of the delivery phase on the demands of other users in the perspective of user k. The privacy constraint in (10) (zero information leakage) corresponds to perfect secrecy in an information-theoretic sense (see [32, Chapter 22]).

Since Z_k is independent of $\mathbb{D}$, the privacy constraint in (10) can be also written as

$$I(\mathbb{D}_{\backslash\{k\}}; X|Z_k, \mathbf{d}_k) = 0, \ \forall k \in [\mathsf{K}]. \text{ (Privacy)} \tag{11}$$

Objective. By the constraint of privacy, we can see that the transmitted loads for different demand matrices should be the same; otherwise, the transmitted load which can be counted by each user will reveal information about the users demands. The memory-load tradeoff (M, R) is

said to be achievable for the memory constraint M, if there exist a two-phase private caching scheme as defined above such that all possible demand matrices can be delivered with load at most R while the decodability and privacy constraints in (9) and (11) are satisfied. The objective is to determine, for a fixed $M \in [0, N]$, the minimum load $R^\star$.

Notice that in the rest of the paper, when we introduce achievable schemes, we directly provide the construction of the payloads and skip the description on their metadatas.

B. MAN Coded Caching Scheme

In the following, we recall the MAN shared-link caching scheme proposed in [2] and show this scheme cannot preserve the privacy of the users demands. We first focus on $L = 1$, i.e., each user requests one file.

Placement Phase. Let $M = Nt'/K$ where $t' \in [0 : K]$. Each file F_i where $i \in [N]$ is divide into $\binom{K}{t'}$ non-overlapping and equal-length subfiles, $F_i = \{F_{i,\mathcal{W}} : \mathcal{W} \subseteq [K], |\mathcal{W}| = t'\}$, while each user $k \in [K]$ caches $F_{i,\mathcal{W}}$ where $k \in \mathcal{W}$. In other words,

$$C_k = \{F_{i,\mathcal{W}} : i \in [N], \mathcal{W} \subseteq [K], |\mathcal{W}| = t', k \in \mathcal{W}\}, \forall k \in [K]. \tag{12}$$

Hence, each user caches $NB\frac{\binom{K-1}{t'-1}}{\binom{K}{t'}} = MB$ bits.

Delivery Phase. For each $\mathcal{S} \subseteq [K]$ where $|\mathcal{S}| = t' + 1$, the server generates an MAN multicast message

$$X_{\mathcal{S}} = \underset{k \in \mathcal{S}}{\oplus} F_{d_{k,1}, \mathcal{S} \setminus \{k\}}. \tag{13}$$

The server transmits $X = \left(X_{\mathcal{S}} : \mathcal{S} \subseteq [K], |\mathcal{S}| = t'+1\right)$. **In this paper, we define the composition of an XOR message of subfiles (or MDS coded symbols) as the containing subfiles (or MDS coded symbols) in this message.** It can be seen that in the composition of $X_{\mathcal{S}}$, each user in $\mathcal{S}$ caches all subfiles except $F_{d_{k,1}, \mathcal{S} \setminus \{k\}}$ such that it can recover $F_{d_{k,1}, \mathcal{S} \setminus \{k\}}$. Considering all $\mathcal{S} \subseteq [K]$ where $|\mathcal{S}| = t' + 1$, each user can recover its desired file, i.e., the decodability constraint in (9) is satisfied.

When $L > 1$, the transmission is divided into L rounds, where in each round we serve one demand of each user. By using the above MAN delivery scheme by L times, the achieved load by the L-round MAN coded caching scheme is as follows,

$$(M_{\text{MAN}}, R_{\text{MAN}}) = \left(\frac{Nt'}{K}, L\frac{K - t'}{t' + 1}\right), \forall t' \in [0 : N]. \tag{14}$$

For other memory sizes, we can take the lower convex envelope of the corner points in (14).

However, consider one $\mathcal{S} \subseteq [\mathsf{K}]$ where $|\mathcal{S}| = t' + 1$, each user knows the metadata of each subfile in $X_{\mathcal{S}}$. Hence, it knows the composition of $X_{\mathcal{S}}$, i.e, it knows the union set of the demanded files by users in $\mathcal{S}$ is $\cup_{k \in \mathcal{S}} \{d_{k,1}\}$, which contradicts the privacy constraint in (11). Even if we hide the identity of the intended users of each multicast messages (i.e., each user $k \in \mathcal{S}$ does not know that $X_{\mathcal{S}}$ is useful to users in $\mathcal{S} \setminus \{k\}$), the composition of the set of all multicast messages is not symmetric over all the N files if the number of users demanding each file is not the same.

III. MAIN RESULTS

In this section, we list the proposed results of this paper for the considered problem described in Section II-A, and then provide some numerical evaluations.

We first provide a baseline scheme, which trivially uses uncoded caching to let each user recover the whole library.

Theorem 1 (Baseline Scheme)**.** *For the* $(\mathsf{K}, \mathsf{N}, \mathsf{M})$ *shared-link caching system with private demands,* $\mathsf{R}^\star$ *is upper bounded by*

$$\mathsf{R}^\star \leq \mathsf{R}_{base} = \mathsf{N} - \mathsf{M}. \tag{15}$$

$\square$

Proof: Placement Phase. Each user caches the same MB/N bits of each file F_i where $i \in [\mathsf{N}]$. We denote the cached part of F_i by F_i^{c} and the uncached part by F_i^{u}. Since users have the same cached content, each user knows the cached content of other users.

Delivery Phase. The server transmits $X = \{F_i^{\mathrm{u}} : i \in [\mathsf{N}]\}$. For the decodability, each user has received uncached part of each file in the library, which includes its desired files. For the privacy, since we transmit the the uncached parts of all files, each user cannot know which files among them are desired by other users. Hence, the privacy of the users demands is preserved.

Performance. The normalized length of the uncached part of each file is $1 - \frac{\mathsf{M}}{\mathsf{N}}$. Hence, the achieved load is $\mathsf{N} \left(1 - \frac{\mathsf{M}}{\mathsf{N}}\right) = \mathsf{N} - \mathsf{M}$ as in Theorem 1. ■

In order to use the coded caching strategy while preserving the privacy of the users' demands, we aim to design private caching schemes such that the composition of the set of all multicast messages is symmetric over all the N files from the viewpoint of each user. For this purpose, with a novel idea of private placement precoding summarized in Remark 2 (which makes the cached

(resp. uncached) bits from each file equivalent from the viewpoint of each user), we propose two private caching schemes, the *virtual-user* scheme and the *MDS-based scheme* scheme, based on two different strategies, respectively. The main ingredients of the two schemes are as follows.

1) Virtual-user scheme. Since in the library there are N files while each user demands L among them (i.e., the demand set of each user contains L files), it can be seen that there are totally $\binom{N}{L}$ possibilities of demand sets, each of which is requested by at most K users. Hence, we can generate $\binom{N}{L}K - K$ virtual users such that the system contains totally $\binom{N}{L}K$ effective users (i.e., real or virtual users) and each possible demand set is requested by exactly K effective users. We then use the MAN delivery scheme over these $\binom{N}{L}K$ effective users. **To conclude, the strategy is that even if the composition of each multicast message is not symmetric over all the N files, with the fact that the number of effective users demanding each file is identical, we let the composition of the set of all multicast messages be symmetric over all the N files from the viewpoint of each user.**

2) MDS-based scheme. **Different from the first strategy, the second strategy is letting each multicast message be symmetric over all the N files from the viewpoint of each user.** More precisely, with a novel private MDS-coded cache placement, we generate symmetric multicast messages in the delivery phase, such that each multicast message (assumed to be useful to users in $\mathcal{S}$) contains one MDS-coded symbol from each file, where each user $k \in \mathcal{S}$ caches all MDS-coded symbols from the files which it does not require. As a result, the composition of the multicast messages is equivalent for different demands from the viewpoint of each user. With some careful design, there is no MDS-coded symbol appearing in two multicast messages, which makes the composition of the set of all multicast messages also symmetric over all the N files.

The achieved load of the virtual user scheme is given in the following, whose proof could be found in Section IV-A.

Theorem 2 (Virtual-user scheme). *For the* (K, N, M, L) *shared-link caching system with private demands,* $R^\star$ *is upper bounded by* R_v, *where the memory-load tradeoff* (M, R_v) *is the lower convex envelope of* $(0, N)$ *and the following memory-load pairs*

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

$\square$

Notice that the idea to introduce virtual user to hide the demands of the real users was originally proposed in [22]. [22] focuses on the case of single request (i.e., $L = 1$) and generates an arbitrary number of virtual users each of whom randomly demands one file, which cannot guarantee the information-theoretic privacy constraint in (10) even if the number of virtual users goes to infinity. Instead, we propose rigorous code constructions on novel private placement and delivery phases, such that the information-theoretic privacy constraint in (10) holds. In the proposed virtual-user scheme we introduce a finite and fixed number of virtual users which depends on the system parameters, and a determinate scenario to choose one demand set for each virtual user.

Compared to the existing converse bound in [4], [5], [8] for the shared-link caching model without privacy, we have the following order optimality results of the virtual-user scheme which will be proved in Appendix A-A.

Theorem 3 (Order Optimality)**.** *For the* (K, N, M, L) *shared-link caching system with private demands,*

- *if* $L = 1$*, the virtual-user scheme in Theorem 2 is order optimal within a factor of* 8 *when* $N \leq K$*, and of* 4 *when* $N > K$ *and* $M \geq N/K$*;*
- *if* $L > 1$*, the virtual-user scheme in Theorem 2 is order optimal within a factor of* 22 *when* $N \leq LK$*, or when* $N > LK$ *and* $M \geq N/K$*.*

$\square$

Intuitively, the order optimality results arise from the fact that introducing virtual users does not increase much load when the memory size is not small (a similar observation was originally pointed out in [22]).

The virtual-user scheme in Theorem 2 contains $\binom{N}{L}K$ effective users, and generate a subfile of each file which is then cached by effective users in $\mathcal{S}$ for each $\mathcal{S} \subseteq [K]$ where $|\mathcal{S}| = t$. Hence, the needed sub-packetization level is

$$\binom{\binom{N}{L}K}{t} \approx 2^{\binom{N}{L}K\mathcal{H}(M/N)}$$

where $\mathcal{H}(p) = -p\log_2(p) - (1-p)\log_2(1-p)$ is the binary entropy function. Hence, the maximal sub-packetization level of the virtual-user scheme (when $M/N \approx 1/2$) is exponential to $\binom{N}{L}K$ (i.e., $\mathcal{O}\left(2^{\binom{N}{L}K}\right)$), which is much higher than the maximal sub-packetization level of the K-user MAN coded caching scheme without virtual users (exponential to K, i.e., $\mathcal{O}\left(2^K\right)$). To

enable the application of the private caching scheme in the practice, it is important to reduce the sub-packetization level (at least the maximal sub-packetization level should not be exponentially larger than the original MAN coded caching scheme). Hence, we propose the MDS-based scheme with sub-packetization level $\mathcal{O}\left(2^K\right)$. The detailed description of the MDS-based scheme and the proof of its achieved load can be found in Sections IV-B and IV-C.

Theorem 4 (MDS-based scheme). *For the* (K, N, M, L) *shared-link caching system with private demands,* $R^\star$ *is upper bounded by* R_m*, where the memory-load tradeoff* (M, R_m) *is the lower convex envelope of* $(0, N)$*, and the following memory-load pairs*

$$\left(N \frac{2^{K-1}}{2^{K-1} + \binom{K-1}{t} + \binom{K-1}{t+1} + \cdots + \binom{K-1}{K-1}}, L \frac{2^K - \binom{K}{0} - \binom{K}{1} - \cdots - \binom{K}{t}}{2^{K-1} + \binom{K-1}{t} + \binom{K-1}{t+1} + \cdots + \binom{K-1}{K-1}} \right), \ \forall t \in [0:K], \tag{17}$$

and $\left(\dfrac{2K-1}{2K} N, \dfrac{L}{2K} \right)$. $\tag{18}$

$\square$

Compared to the existing converse bound in [4], [5], [8] for the shared-link caching model without privacy, the virtual-user scheme and the MDS-based scheme have the same order optimality results when $M \geq N/2$, which will be proved in Appendix A-B.

Theorem 5 (Order Optimality). *For the* (K, N, M, L) *shared-link caching system with private demands, when* $M \geq N/2$*, both of* R_v *and* R_m *are order optimal within a factor of* 2. $\square$

By comparing the achievable bounds in (17) (letting $t = K - 1$) and (18), with the converse bound for the MAN shared-link caching model with multiple requests in [33, Theorem 1] (letting $s = 1$), we have the following exact optimality result.

Theorem 6 (Exact Optimality). *For the* (K, N, M, L) *shared-link caching system with private demands where* $M \geq \min\left\{ \frac{2K-1}{2K}, \frac{2^{K-1}}{2^{K-1}+1} \right\} N$*, we have*

$$R^\star = R_m = L\left(1 - \frac{M}{N}\right). \tag{19}$$

$\square$

It can be seen that for the considered large memory size regime in Theorem 6, our proposed schemes can maintain the exact optimality for the shared-link caching model with multiple

(a) $(K, N, L) = (10, 20, 1)$.
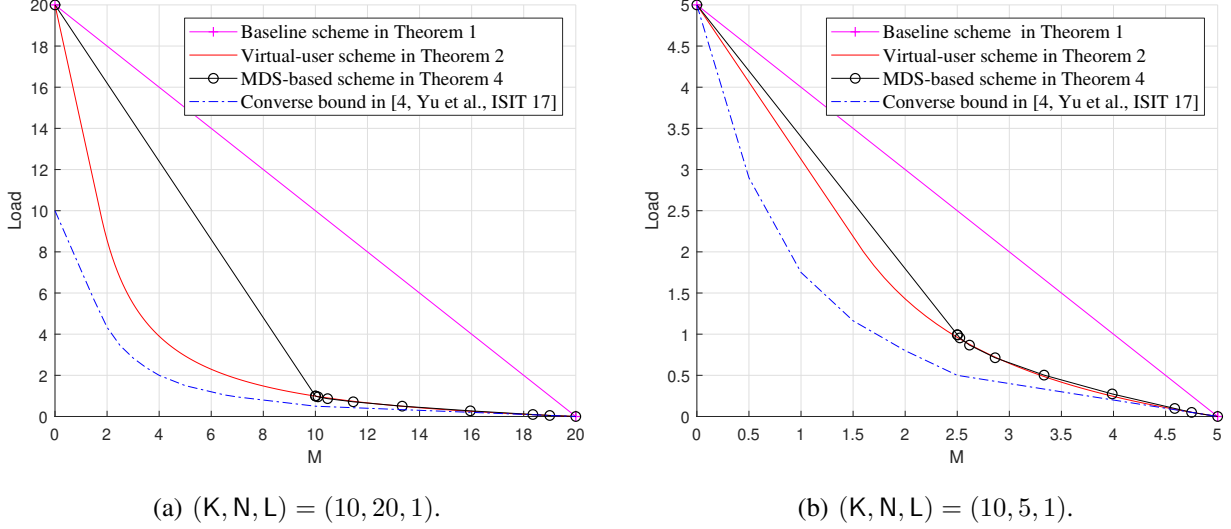
(b) $(K, N, L) = (10, 5, 1)$.

Figure 2: (M, R) tradeoff for the (K, N, M, L) shared-link caching system with private demands.

requests, while preserving the privacy of the users demands. Notice that for this purpose, the virtual-user scheme needs the memory size no less than $\frac{\binom{N}{L}K - 1}{\binom{N}{L}K}N$.

From Theorems 3 and 5, the only open case, where the multiplicative gaps between the proposed schemes and the existing converse bounds for the shared-link caching model without privacy constraint are not constant, is when $N < LK$ and $M < N/K$.

Finally, we provide numerical evaluations of the proposed private caching schemes for the (K, N, M, L) shared-link caching system with private demands. In Fig. 2 we let $L = 1$ and use the converse bound in [4] for the shared-link caching model with single request, as the converse bound in our problem. In Fig. 2a, we let $(K, N) = (10, 20)$ and in Fig. 2b we let $(K, N) = (10, 5)$. Both of the figures show that the virtual-user scheme and the MDS-based scheme outperform the baseline scheme. When $M < N/2$, it can be seen that the achieved load by the virtual-user scheme is lower than the MDS-based scheme. In addition, when $M \geq N/2$, the achieved loads by the virtual-user scheme and the MDS-based scheme are close; in this regime, none of them always has the lower load than the other.

IV. CODED CACHING WITH PRIVATE DEMANDS

A. Proof of Theorem 2

In the following, we describe the virtual-user scheme which achieves the memory-load tradeoff in (16). We focus on one $t \in \left[\binom{N}{L}K\right]$. We define that $U := \binom{N}{L}K$.

Placement Phase. Each file F_i where $i \in [\mathsf{N}]$ is divided into $\binom{\mathsf{U}}{t}$ non-overlapping and equal-length pieces, denoted by $S^i_1, \ldots, S^i_{\binom{\mathsf{U}}{t}}$, where each piece has $\frac{\mathsf{B}}{\binom{\mathsf{U}}{t}}$ bits. We randomly generate a permutation of $\binom{\mathsf{U}}{t}$, denoted by $\mathbf{p}_i = (p_{i,1}, \ldots, p_{i,\binom{\mathsf{U}}{t}})$, independently and uniformly over the set of all possible permutations. We sort all sets $\mathcal{W} \subseteq [\mathsf{U}]$ where $|\mathcal{W}| = t$, in a lexicographic order, denoted by $\mathcal{W}(1), \ldots, \mathcal{W}\left(\binom{\mathsf{U}}{t}\right)$. For each $j \in \left[\binom{\mathsf{U}}{t}\right]$, we generate a subfile

$$f_{i,\mathcal{W}(j)} = S^i_{p_{i,j}}. \tag{20}$$

Each user $k \in [\mathsf{K}]$ caches $f_{i,\mathcal{W}}$ where $\mathcal{W} \subseteq [\mathsf{U}]$, $|\mathcal{W}| = t$, and $k \in \mathcal{W}$. Hence, each user caches $\binom{\mathsf{U}-1}{t-1}$ subfiles of each file, and thus it totally caches $\frac{\binom{\mathsf{U}-1}{t-1}}{\binom{\mathsf{U}}{t}}\mathsf{NB} = \frac{t}{\mathsf{U}}\mathsf{NB} = \mathsf{MB}$ bits, satisfying the memory size constraint in (16).

For each subfile of F_i cached by user $k \in [\mathsf{K}]$, since the random permutation $\mathbf{p}_i$ is unknown to user k, it does not know the other users who also cache it. Hence, each cached subfile of F_i is equivalent from the viewpoint of user k. Similarly, each uncached subfile of F_i is also equivalent from the viewpoint of user k. Hence, from the viewpoint of user $k \in [\mathsf{K}]$, each cached subfile of F_i is equivalent from the viewpoint of user k, while each uncached subfile of F_i is also equivalent.

Delivery Phase for $\mathbb{D}$. Recall that for one possible demand vector by one user $\mathbf{d} := (d_1, d_2, \ldots, d_\mathsf{L})$, we should have $1 \le d_1 < d_2 < \cdots < d_\mathsf{L} \le \mathsf{N}$. Hence, there are totally $\binom{\mathsf{N}}{\mathsf{L}}$ possible demand vectors, denoted by $\mathbf{d}^1, \ldots, \mathbf{d}^{\binom{\mathsf{N}}{\mathsf{L}}}$. We define that

$$n_j := |\{k \in [\mathsf{K}] : \mathbf{d}_k = \mathbf{d}^j\}| \tag{21}$$

where $j \in \left[\binom{\mathsf{N}}{\mathsf{L}}\right]$, representing the number of real users demanding the demand vector $\mathbf{d}^j$. We then allocate one demand vector to each of the $\mathsf{U} - \mathsf{K}$ virtual users as follows. For each $j \in \left[\binom{\mathsf{N}}{\mathsf{L}}\right]$, we let $\mathbf{d}_{1+j\mathsf{K}-\sum_{q\in[j-1]}n_q} = \cdots = \mathbf{d}_{(j+1)\mathsf{K}-\sum_{q\in[j]}n_q} = \mathbf{d}^j$. For example, when $j = 1$, we let $\mathbf{d}_{\mathsf{K}+1} = \cdots = \mathbf{d}_{2\mathsf{K}-n_1} = \mathbf{d}^1$; when $j = 2$, we let $\mathbf{d}_{2\mathsf{K}-n_1+1} = \cdots = \mathbf{d}_{3\mathsf{K}-n_1-n_2} = \mathbf{d}^2$. Hence, by this way, each possible demand vector is requested by K effective (real or virtual) users.

Recall for any $k \in [\mathsf{U}]$, we define $\mathbf{d}_k = (d_{k,1}, \ldots, d_{k,\mathsf{L}})$. In addition, we define $\mathbb{G}_{a \times b}$ as the $a \times b$ parity-check matrix of the $[b, b-a, a+1]$ MDS code (or an $a \times n$ Cauchy matrix) such that each a columns are linearly independent (see [34]). For each set $\mathcal{S} \subseteq [\mathsf{U}]$ where $|\mathcal{S}| = t+1$, we generate

$$X_\mathcal{S} = \mathbb{G}_{\mathsf{L} \times \mathsf{L}(t+1)} \left[\mathbf{f}_{k_1, \mathcal{S} \setminus \{k_1\}}; \mathbf{f}_{k_2, \mathcal{S} \setminus \{k_2\}}; \ldots; \mathbf{f}_{k_{t+1}, \mathcal{S} \setminus \{k_{t+1}\}}\right]. \tag{22}$$

containing L combinations where $(k_1, \ldots, k_{t+1})$ is a random permutation of $\mathcal{S}$ independently and uniformly over the set of all possible permutations, and we define

$$\mathbf{f}_{k_j, \mathcal{S} \setminus \{k_j\}} := [f_{d_{k_j}, 1, \mathcal{S} \setminus \{k_j\}}; f_{d_{k_j}, 2, \mathcal{S} \setminus \{k_j\}}; \ldots; f_{d_{k_j}, \mathsf{L}, \mathcal{S} \setminus \{k_j\}}], \ \forall j \in [t+1]. \tag{23}$$

Finally, we randomly generate a permutation of $\left[\binom{\mathsf{U}}{t+1} \right]$, denoted by $\mathbf{q} = (q_1, \ldots, q_{\binom{\mathsf{U}}{t+1}})$, independently and uniformly over the set of all possible permutations. We sort all sets $\mathcal{S} \subseteq [\mathsf{U}]$ where $|\mathcal{S}| = t+1$, in a lexicographic order, denoted by $\mathcal{S}(1), \ldots, \mathcal{S}\left(\binom{\mathsf{U}}{t+1}\right)$. The server transmit

$$X = \left(X_{\mathcal{S}(q_1)}, \ldots, X_{\mathcal{S}\left(q_{\binom{\mathsf{U}}{t+1}} \right)} \right). \tag{24}$$

Decodability. We focus on user $k \in [\mathsf{K}]$. From the metadata in X (i.e., $\mathscr{M}(P)$), for each $j \in \left[\binom{\mathsf{U}}{t+1} \right]$, user $k \in [\mathsf{K}]$ checks $X_{\mathcal{S}(q_j)}$. If $X_{\mathcal{S}(q_j)}$ contains $\mathsf{L}t$ cached subfiles from the files not requested by user k, and L subfiles from the files requested by user k, user k knows $X_{\mathcal{S}(q_j)}$ is useful to it and then decodes the L requested subfiles from the L linear combinations in $X_{\mathcal{S}(q_j)}$, because any L columns in $\mathbb{G}_{\mathsf{L} \times \mathsf{L}(t+1)}$ are linearly independent.

After considering all transmitted packets in X, user $k \in [\mathsf{K}]$ can recover all requested subfiles to reconstruct its requested files.

Privacy. By the symmetric construction, from the viewpoint of each user $k \in [\mathsf{K}]$, for any demand matrix where user k demands $\mathbf{d}_k$, there are always K effective users demanding each possible demand vector. In addition, since the placement permutations (i.e., $\mathbf{p}_i$ where $i \in [\mathsf{N}]$) is unknown to user k, the cached content of each of the other $\mathsf{U} - 1$ effective users is equivalent from the viewpoint of user k. Hence, the composition of X is totally equivalent for different demand matrices from the viewpoint of each user $k \in [\mathsf{K}]$. In other words, given $\mathbf{d}_k$ and Z_k, it can be seen that X is independent of $\mathbb{D}$. Thus the proposed scheme is information-theoretically private.

Performance. For any demand matrix, we transmit $\binom{\mathsf{U}}{t+1}$ messages, each of which contains $\frac{\mathsf{LB}}{\binom{\mathsf{U}}{t}}$ bits. Hence, the achieved load is $\mathsf{L} \frac{\binom{\mathsf{U}}{t+1}}{\binom{\mathsf{U}}{t}} = \mathsf{L} \frac{\mathsf{U}-t}{t+1}$, as shown in (16). The sub-packetization level is $\binom{\mathsf{U}}{t}$.

B. Proof of (17)

In the following we introduce the MDS-based scheme to achieve (17). We first use a more complicated example than the toy example in Section I-C to highlight more insights.

Example 1 ($K = 3$, $N = 6$, $M = 3$, $L = 2$)**.** Consider a $(K, N, M, L) = (3, 6, 3, 2)$ shared-link caching problem with private demands.

Placement Phase. From (17), we can compute $t = 0$ in this example. Each file F_i where $i \in [N]$ is divided into $\binom{K}{0} + \cdots + \binom{K}{K} = 2^K = 8$ non-overlapping and equal-length pieces, denoted by $S_1^i, \ldots, S_{2^K}^i$, each of which contains $\frac{B}{8}$ bits. In this example where $t = 0$, we do not need the MDS precoding in the placement, which is necessary for $t > 1$ and will be clarified in the next example. We randomly generate a permutation of $[2^K]$, denoted by $\mathbf{p}_i = (p_{i,1}, \ldots, p_{i,2^K})$, independently and uniformly over the set of all possible permutations. We then assign each piece to a subfile according to $\mathbf{p}_i$ as follows,

$$f_{i,\emptyset} = S_{p_{i,1}}^i, \ \ f_{i,\{1\}} = S_{p_{i,2}}^i, \ \ f_{i,\{1,2\}} = S_{p_{i,3}}^i, \ \ f_{i,\{1,2,3\}} = S_{p_{i,4}}^i,$$

$$f_{i,\{1,3\}} = S_{p_{i,5}}^i, \ \ f_{i,\{2\}} = S_{p_{i,6}}^i, \ \ f_{i,\{2,3\}} = S_{p_{i,7}}^i, \ \ f_{i,\{3\}} = S_{p_{i,8}}^i. \tag{25}$$

Each user $k \in [K]$ caches $f_{i,\mathcal{W}}$ if $k \in \mathcal{W}$, i.e., the cached contents of the three users for each file F_i are as follows:

- User 1 stores $f_{i,\{1\}}$, $f_{i,\{1,2\}}$, $f_{i,\{1,3\}}$, and $f_{i,\{1,2,3\}}$.
- User 2 stores $f_{i,\{2\}}$, $f_{i,\{1,2\}}$, $f_{i,\{2,3\}}$, and $f_{i,\{1,2,3\}}$.
- User 3 stores $f_{i,\{3\}}$, $f_{i,\{1,3\}}$, $f_{i,\{2,3\}}$, and $f_{i,\{1,2,3\}}$.

In addition, for each subfile of F_i cached by user $k \in [K]$, since the random permutation $\mathbf{p}_i$ is unknown to user k, it does not know the other users who also cache it. Hence, each cached subfile of F_i is equivalent from the viewpoint of user k. Similarly, each uncached subfile of F_i is also equivalent from the viewpoint of user k.

Since each user caches 4 subfiles (each of which has $B/8$ bits) for each file in its cache, it totally caches $N\frac{4B}{8} = 3B = MB$ bits satisfying the memory size constraint.

For the delivery phase, we do not consider all possible non-equivalent demand configurations, for the sake of brevity. Instead, we give two explicit examples of the construction of the delivery phase and then extract some general properties that demonstrate the privacy.

Delivery Phase for $\mathbb{D} = [1, 2; 3, 4; 5, 6]$. For this demand matrix, user 1 demands F_1 and F_2, user 2 demands F_3 and F_4, and user 3 demands F_5 and F_6. For each file F_i where $i \in [N]$, we define

$$\mathcal{Q}_i := \{k \in [K] : i \in \mathbf{d}_k\}, \tag{26}$$

as the set of users demanding F_i. For $\mathbb{D} = [1, 2; 3, 4; 5, 6]$, we have $\mathcal{Q}_1 = \mathcal{Q}_2 = \{1\}$, $\mathcal{Q}_3 = \mathcal{Q}_4 = \{2\}$, and $\mathcal{Q}_5 = \mathcal{Q}_6 = \{3\}$.

For each subset $\mathcal{S} \subseteq [\mathsf{K}]$ where $|\mathcal{S}| \geq t + 1 = 1$, we generate a multicast message $X_\mathcal{S}$ which is useful to the users $\mathcal{S}$. Our purpose is to let $X_\mathcal{S}$ be $\mathsf{L} = 2$ linear combinations of N subfiles, where each file has one subfile in $X_\mathcal{S}$ and each user in $\mathcal{S}$ caches $\mathsf{N} - \mathsf{L}$ subfiles from the files which it does not request. In addition, the $\mathsf{L} = 2$ linear combinations are generated by $\mathbb{G}_{\mathsf{L} \times \mathsf{N}}$ where each L columns are linearly independent, such that each user in $\mathcal{S}$ can recover the L uncached subfiles. For each file F_i where $i \in [\mathsf{N}]$, the subfile of F_i in $X_\mathcal{S}$ is $f_{i, \mathcal{S} \cup \mathcal{Q}_i \setminus (\mathcal{S} \cap \mathcal{Q}_i)}$ (the motivation of this construction will be explained in Remark 1), which is cached by each user in $\mathcal{S}$ not requesting F_i, and not cached by each user in $\mathcal{S}$ requesting F_i.

We first consider $\mathcal{S} = \{1\}$, which only contains one user. We have

$$X_{\{1\}} = \mathbb{G}_{2 \times 6} \left[f_{1,\emptyset}; f_{2,\emptyset}; f_{3,\{1,2\}}; f_{4,\{1,2\}}; f_{5,\{1,3\}}; f_{6,\{1,3\}} \right]. \tag{27}$$

From $X_{\{1\}}$, user 1 caches all except $f_{1,\emptyset}$ and $f_{2,\emptyset}$, such that it can recover those two subfiles in $X_{\{1\}}$ (recall each two columns of $\mathbb{G}_{2 \times 6}$ are linearly independent). Similarly, we have

$$X_{\{2\}} = \mathbb{G}_{2 \times 6} \left[f_{1,\{1,2\}}; f_{2,\{1,2\}}; f_{3,\emptyset}; f_{4,\emptyset}; f_{5,\{2,3\}}; f_{6,\{2,3\}} \right], \tag{28}$$

$$X_{\{3\}} = \mathbb{G}_{2 \times 6} \left[f_{1,\{1,3\}}; f_{2,\{1,3\}}; f_{3,\{2,3\}}; f_{4,\{2,3\}}; f_{5,\emptyset}; f_{6,\emptyset} \right]. \tag{29}$$

We then consider $\mathcal{S} = \{1, 2\}$, which contains two users. We have

$$X_{\{1,2\}} = \mathbb{G}_{2 \times 6} \left[f_{1,\{2\}}; f_{2,\{2\}}; f_{3,\{1\}}; f_{4,\{1\}}; f_{5,\{1,2,3\}}; f_{6,\{1,2,3\}} \right]. \tag{30}$$

From $X_{\{1,2\}}$, user 1 caches all except $f_{1,\{2\}}$ and $f_{2,\{2\}}$, such that it can recover those two subfiles in $X_{\{1,2\}}$. In addition, user 2 can recover $f_{3,\{1\}}$ and $f_{4,\{1\}}$ from $X_{\{1,2\}}$. Similarly, we have

$$X_{\{1,3\}} = \mathbb{G}_{2 \times 6} \left[f_{1,\{3\}}; f_{2,\{3\}}; f_{3,\{1,2,3\}}; f_{4,\{1,2,3\}}; f_{5,\{1\}}; f_{6,\{1\}} \right], \tag{31}$$

$$X_{\{2,3\}} = \mathbb{G}_{2 \times 6} \left[f_{1,\{1,2,3\}}; f_{2,\{1,2,3\}}; f_{3,\{3\}}; f_{4,\{3\}}; f_{5,\{2\}}; f_{6,\{2\}} \right]. \tag{32}$$

Finally we consider $\mathcal{S} = \{1, 2, 3\}$, which contains three users. We have

$$X_{\{1,2,3\}} = \mathbb{G}_{2 \times 6} \left[f_{1,\{2,3\}}; f_{2,\{2,3\}}; f_{3,\{1,3\}}; f_{4,\{1,3\}}; f_{5,\{2,3\}}; f_{6,\{2,3\}} \right]. \tag{33}$$

From $X_{\{1,2,3\}}$, user 1 caches all except $f_{1,\{2,3\}}$ and $f_{2,\{2,3\}}$, such that it can recover those two subfiles in $X_{\{1,2,3\}}$. In addition, user 2 can recover $f_{3,\{1,3\}}$ and $f_{4,\{1,3\}}$ while user 3 can recover $f_{5,\{1,2\}}$ and $f_{6,\{1,2\}}$.

Hence, the server transmits $X = (X_{\mathcal{S}} : \mathcal{S} \subseteq [\mathsf{K}], |\mathcal{S}| \in [3])$, such that each user can recover its desired files in the delivery phase. For the privacy constraint in (11), we let then focus on the demand matrix $\mathbb{D} = [1, 2; 1, 3; 1, 4]$, and show the compositions of the received multicast messages by each user are equivalent from its viewpoint to the ones for the demand matrix $\mathbb{D} = [1, 2; 3, 4; 5, 6]$.

Delivery Phase for $\mathbb{D} = [1, 2; 1, 3; 1, 4]$. For $\mathbb{D} = [1, 2; 1, 3; 1, 4]$, we have $\mathcal{Q}_1 = \{1, 2, 3\}$, $\mathcal{Q}_2 = \{1\}$, $\mathcal{Q}_3 = \{2\}$, $\mathcal{Q}_4 = \{3\}$, and $\mathcal{Q}_5 = \mathcal{Q}_6 = \emptyset$. From the same way to construct multicast messages as described above, for $\mathbb{D} = [1, 2; 1, 3; 1, 4]$ we have

$$X_{\{1\}} = \mathbb{G}_{2 \times 6} \, [f_{1,\{2,3\}}; f_{2,\emptyset}; f_{3,\{1,2\}}; f_{4,\{1,3\}}; f_{5,\{1\}}; f_{6,\{1\}}], \tag{34}$$

$$X_{\{2\}} = \mathbb{G}_{2 \times 6} \, [f_{1,\{1,3\}}; f_{2,\{1,2\}}; f_{3,\emptyset}; f_{4,\{2,3\}}; f_{5,\{2\}}; f_{6,\{2\}}], \tag{35}$$

$$X_{\{3\}} = \mathbb{G}_{2 \times 6} \, [f_{1,\{1,2\}}; f_{2,\{1,3\}}; f_{3,\{2,3\}}; f_{4,\emptyset}; f_{5,\{3\}}; f_{6,\{3\}}], \tag{36}$$

$$X_{\{1,2\}} = \mathbb{G}_{2 \times 6} \, [f_{1,\{3\}}; f_{2,\{2\}}; f_{3,\{1\}}; f_{4,\{1,2,3\}}; f_{5,\{1,2\}}; f_{6,\{1,2\}}], \tag{37}$$

$$X_{\{1,3\}} = \mathbb{G}_{2 \times 6} \, [f_{1,\{2\}}; f_{2,\{3\}}; f_{3,\{1,2,3\}}; f_{4,\{1\}}; f_{5,\{1,3\}}; f_{6,\{1,3\}}], \tag{38}$$

$$X_{\{2,3\}} = \mathbb{G}_{2 \times 6} \, [f_{1,\{1\}}; f_{2,\{1,2,3\}}; f_{3,\{3\}}; f_{4,\{2\}}; f_{5,\{2,3\}}; f_{6,\{2,3\}}], \tag{39}$$

$$X_{\{1,2,3\}} = \mathbb{G}_{2 \times 6} \, [f_{1,\emptyset}; f_{2,\{2,3\}}; f_{3,\{1,3\}}; f_{4,\{1,2\}}; f_{5,\{1,2,3\}}; f_{6,\{1,2,3\}}], \tag{40}$$

and let the server transmit $X = (X_{\mathcal{S}} : \mathcal{S} \subseteq [\mathsf{K}], |\mathcal{S}| \in [3])$.

Privacy. For any demand matrix (we do not list the transmission for all demand matrices for sake of simplicity), we can summarize four common points:

1) for any $i \in [\mathsf{N}]$, each subfile of F_i cached by user 1 is equivalent from the viewpoint of user 1; each subfile of F_i not cached by user 1 is also equivalent from the viewpoint of user 1;

2) there does not exist any subfile appearing in two multicast messages, which ensures both the decodability and privacy.

3) in each of $X_{\{1\}}, X_{\{1,2\}}, X_{\{1,3\}}, X_{\{1,2,3\}}$, there is exactly one subfile of each file. If this subfile is from a file requested by user 1, it is uncached by user 1; otherwise, it is cached by user 1.

4) in each of $X_{\{2\}}, X_{\{3\}}, X_{\{2,3\}}$, there is exactly one subfile of each file. If this subfile is from a file requested by user 1, it is cached by user 1; otherwise, it is uncached by user 1.

Hence, from the viewpoint of user 1, the composition of X (i.e., the subfiles in each XOR multicast message), is symmetric for different demand matrices in which $\mathbf{d}_1 = (1, 2)$. In other

words, knowing $\mathbf{d}_1$ and Z_1, the probability that X is generated for any demand matrix $\mathbb{D}_{\setminus\{1\}}$, is identical. Similarly, for any user in $[\mathsf{K}]$, it cannot get any information about the demands of other users neither. The formal information-theoretic proof on the privacy constraint in (11) of the new private caching scheme can be found in Appendix C.

Performance. For any demand matrix, we transmit $\binom{\mathsf{K}}{1} + \cdots + \binom{\mathsf{K}}{\mathsf{K}} = 2^{\mathsf{K}} - 1 = 7$ multicast messages, each of which contains $\mathsf{L} = 2$ linear combinations of subfiles. Since each subfile has $B/8$ bits, the load in the delivery phase is $14/8 = 1.75$ with sub-packetization level 8. The achieved load by the virtual-user scheme in Theorem 2 is $23/12 \approx 1.92$ with sub-packetization level $2^{\binom{\mathsf{N}}{\mathsf{L}}\mathsf{K}\mathcal{H}(\mathsf{M}/\mathsf{N})} \approx 2.47 \times 10^{13}$. Notice that the achieved load by the baseline scheme is $\mathsf{N} - \mathsf{M} = 3$. In conclusion, the achieved load by the MDS-based scheme is less than the virtual-user scheme, and with a much lower sub-packetization level.

$\square$

Remark 1. *Besides the high-level privacy strategy of the MDS-based scheme introduced in Section III, there is another important construction which makes the MDS-based scheme private.*

In the multicast message $X_{\mathcal{S}}$, there is one subfile from each file. The subfile for the file F_i is $f_{i,\mathcal{S}\cup\mathcal{Q}_i\setminus(\mathcal{S}\cap\mathcal{Q}_i)}$, instead of $f_{i,\mathcal{S}\setminus\mathcal{Q}_i}$, such that there does not exist any subfile appearing in two multicast messages. We assume $f_{i,\mathcal{S}\setminus\mathcal{Q}_i}$ is transmitted in $X_{\mathcal{S}}$. For each demand matrix where users request different files, one subfile appears in at most two multicast messages, e.g., if F_1 is only demanded by user 1, $f_{1,\{3\}}$ appears in $X_{\{3\}}$ and $X_{\{1,3\}}$. However, if F_1 is demanded by both users $1, 2$ and not by user 3, it can be seen that $f_{1,\{3\}}$ appears in $X_{\{1,3\}}$, $X_{\{2,3\}}$, and $X_{\{1,2,3\}}$. Hence, the composition of the multicast messages depends on the users' demands. $\square$

In the following example, we also consider $\mathsf{K} = 3$, $\mathsf{N} = 6$, $\mathsf{L} = 2$, but with $\mathsf{M} = 24/7$ which leads $t = 1$ in (17). For $t \geq 1$, the new private caching scheme needs an MDS precoding in the placement phase.

Example 2 ($\mathsf{K} = 3$, $\mathsf{N} = 6$, $\mathsf{M} = 24/7$, $\mathsf{L} = 2$)**.** From (17), we can compute $t = 1$.

Placement Phase. Each file F_i where $i \in [\mathsf{N}]$ is divided into $2^{\mathsf{K}-1} + \binom{\mathsf{K}-1}{t} + \cdots + \binom{\mathsf{K}-1}{\mathsf{K}-1} = 7$ non-overlapping and equal-length pieces, which are then encoded by a $\left(2^{\mathsf{K}}, 2^{\mathsf{K}-1} + \binom{\mathsf{K}-1}{t} + \cdots + \binom{\mathsf{K}-1}{\mathsf{K}-1}\right) = (8, 7)$ MDS code (the parameters of the MDS code will be explained later).[1] Each MDS coded symbol has $B/7$ bits. By the property of the MDS code, any 7 MDS coded symbols can

---

[1]When $t = 0$, it can be seen that $2^{\mathsf{K}-1} + \binom{\mathsf{K}-1}{t} + \cdots + \binom{\mathsf{K}-1}{\mathsf{K}-1} = 2^{\mathsf{K}}$. So we do not need the MDS precoding.

reconstruct the whole file. The 8 MDS coded symbols of F_i are denoted by $S_1^i, \ldots, S_8^i$. The rest of the placement phase is the same as $t = 0$ in Example 1. More precisely, we randomly generate a permutation of $[2^K]$, denoted by $\mathbf{p}_i = (p_{i,1}, \ldots, p_{i,2^K})$ and assign each MDS coded symbol to a subfile according to $\mathbf{p}_i$ as in (25). Each user $k \in [K]$ caches $f_{i,\mathcal{W}}$ if $k \in \mathcal{W}$. Hence, each user totally caches $\frac{4B}{7}N = \frac{24B}{7} = MB$ bits satisfying the memory size constraint.

Delivery Phase for $\mathbb{D} = [1, 2; 3, 4; 5, 6]$. For each subset $\mathcal{S} \subseteq [K]$ where $|\mathcal{S}| \geq t + 1 = 2$, we let the server transmit $X_{\mathcal{S}}$ with the same construction in (30)-(33). In other words, compared to Example 1 with $t = 0$, we only transmit $X_{\{1,2\}}, X_{\{1,3\}}, X_{\{2,3\}}, X_{\{1,2,3\}}$.

From $X_{\{1,2\}}, X_{\{1,3\}}, X_{\{1,2,3\}}$, user 1 can recover 3 MDS coded symbols for each of its desired files. Since it caches $2^{K-1} = 4$ MDS coded symbols for each file, it can recover each of its desired files by the $4 + 3 = 7$ MDS coded symbols, and thus it can recover its desired files.

In short, the $X_{\mathcal{S}}$'s where $0 < |\mathcal{S}| < t + 1$ are not transmitted in the delivery phase and thus each user cannot recover all subfiles of its desired files. Hence, we need the MDS precoding for $t \geq 1$.

Privacy. By the same reason as Example 1, the new private scheme for $t = 1$ can also satisfy the privacy constraint.

Performance. For any demand matrix, we transmit 4 multicast messages, each of which contains $L = 2$ linear combinations of subfiles. Since each subfile has $B/7$ bits, the load in the delivery phase is $8/7 \approx 1.14$ with sub-packetization level 8. The achieved load by the virtual-user scheme is $3550/2457 \approx 1.44$ with sub-packetization level $2^{\binom{N}{L}K\mathcal{H}(M/N)} \approx 2.22 \times 10^{13}$. Notice that the load achieved by the baseline scheme is $18/7 \approx 2.57$. As in Example 1, in this example the MDS-based scheme has a lower load and a much lower sub-packetzation level compared to the virtual-user scheme. $\qquad\square$

We are now ready to generalize Examples 1 and 2. We focus on the memory size

$$M = \frac{2^{K-1}}{2^{K-1} + \binom{K-1}{t} + \binom{K-1}{t+1} + \cdots + \binom{K-1}{K-1}} N,$$

where $t \in [0 : K - 1]$. Notice that if $t = K$, we have $M = N$ and each user can store the whole library in its cache, such that the server needs not to transmit any packet in the delivery phase.

Placement Phase. Each file F_i where $i \in [N]$ is divided into $2^{K-1} + \binom{K-1}{t} + \cdots + \binom{K-1}{K-1}$ non-overlapping and equal-length pieces, which are then encoded by a $\left(2^K, 2^{K-1} + \binom{K-1}{t} + \cdots + \binom{K-1}{K-1}\right)$ MDS code. Each MDS coded symbol has $\frac{B}{2^{K-1} + \binom{K-1}{t} + \cdots + \binom{K-1}{K-1}}$ bits, and the MDS coded symbols

of F_i is denoted by $S_1^i, \ldots, S_{2^{\mathsf{K}}}^i$. We randomly generate a permutation of $[2^{\mathsf{K}}]$, denoted by $\mathbf{p}_i = (p_{i,1}, \ldots, p_{i,2^{\mathsf{K}}})$, independently and uniformly over the set of all possible permutations. Recall that $\mathrm{Pow}(a, j)$ denotes the jth set in the power set of $[a]$ with a lexicographic order. For each $j \in [2^{\mathsf{K}}]$, we generate one subfile

$$f_{i,\mathrm{Pow}(\mathsf{K},j)} := S_{p_{i,j}}^i. \tag{41}$$

Any $2^{\mathsf{K}-1} + \binom{\mathsf{K}-1}{t} + \cdots + \binom{\mathsf{K}-1}{\mathsf{K}-1}$ subfiles of F_i can reconstruct F_i. For each $\mathcal{W} \subseteq [\mathsf{K}]$, user $k \in [\mathsf{K}]$ caches $f_{i,\mathcal{W}}$ if $k \in \mathcal{W}$. It can be seen that each user caches $2^{\mathsf{K}-1}$ subfiles of each file. Hence, each user totally caches $\frac{2^{\mathsf{K}-1}}{2^{\mathsf{K}-1} + \binom{\mathsf{K}-1}{t} + \cdots + \binom{\mathsf{K}-1}{\mathsf{K}-1}} \mathsf{N}\mathsf{B} = \mathsf{M}\mathsf{B}$ bits in its cache, satisfying the memory size constraint.

Delivery Phase for $\mathbb{D}$. Recall that $\mathcal{Q}_i$ where $i \in [\mathsf{N}]$ denotes the set of users demanding F_i. For each subset $\mathcal{S} \subseteq [\mathsf{K}]$ where $|\mathcal{S}| \geq t + 1$, the server generates

$$X_{\mathcal{S}} = \mathbb{G}_{\mathsf{L} \times \mathsf{N}} \left[f_{1, \mathcal{S} \cup \mathcal{Q}_1 \setminus (\mathcal{S} \cap \mathcal{Q}_1)}; f_{2, \mathcal{S} \cup \mathcal{Q}_2 \setminus (\mathcal{S} \cap \mathcal{Q}_2)}; \ldots; f_{\mathsf{N}, \mathcal{S} \cup \mathcal{Q}_{\mathsf{N}} \setminus (\mathcal{S} \cap \mathcal{Q}_{\mathsf{N}})} \right]. \tag{42}$$

$X_{\mathcal{S}}$ contains L linear combinations and in $X_{\mathcal{S}}$, each user $k \in \mathcal{S}$ caches all subfiles except $f_{i, \mathcal{S} \cup \mathcal{Q}_i \setminus (\mathcal{S} \cap \mathcal{Q}_i)}$ where $i \in \mathbf{d}_k$. By the property of $\mathbb{G}_{\mathsf{L} \times \mathsf{N}}$ (each L columns are linearly independent), user k can recover $f_{i, \mathcal{S} \cup \mathcal{Q}_i \setminus (\mathcal{S} \cap \mathcal{Q}_i)}$ where $i \in \mathbf{d}_k$. Then we let the server transmit

$$X = (X_{\mathcal{S}} : \mathcal{S} \subseteq [\mathsf{K}], |\mathcal{S}| \geq t + 1). \tag{43}$$

Decodability. We first introduce the following lemma, which will be proved in Appendix B.

Lemma 1. *For any demand matrix* $\mathbb{D} \in \mathscr{D}$, *there is no subfile transmitted in more than one multicast message of the scheme in Section IV-B.*

We focus on user $k \in [\mathsf{K}]$ and file F_i where $i \in \mathbf{d}_k$. For each subset $\mathcal{S} \subseteq [\mathsf{K}]$ where $|\mathcal{S}| \geq t + 1$ and $k \in \mathcal{S}$, user k can recover one uncached subfile of F_i from the multicast message $X_{\mathcal{S}}$. Considering all such subsets, user k can recover $\binom{\mathsf{K}-1}{t} + \cdots + \binom{\mathsf{K}-1}{\mathsf{K}-1}$ uncached subfiles of F_i. By Lemma 1, these subfiles are distinct. Hence, user k can totally obtain $2^{\mathsf{K}-1} + \binom{\mathsf{K}-1}{t} + \cdots + \binom{\mathsf{K}-1}{\mathsf{K}-1}$ subfiles of F_i from the placement and delivery phases, such that it can recover F_i.

Privacy. Let us focus on user k. Intuitively, for each subfile of F_i cached by user k, since the random permutation $\mathbf{p}_i$ is unknown to user k, it does not know the other users who also cache it, and thus each cached subfile of F_i is equivalent from the viewpoint of user k. Similarly, each uncached subfile of F_i is equivalent from the viewpoint of user k. In each multicast message $X_{\mathcal{S}}$ where $\mathcal{S} \subseteq [\mathsf{K}]$ and $|\mathcal{S}| \geq t + 1$,

- when $k \in \mathcal{S}$, there is exactly one subfile of each file. If this subfile is from a file requested by user k, it is uncached by user k; otherwise, it is cached by user k.

- when $k \notin \mathcal{S}$, there is exactly one subfile of each file. If this subfile is from a file requested by user k, it is cached by user k; otherwise, it is uncached by user k.

In addition, by Lemma 1, there does not exist any subfile transmitted in more than one multicast messages. Hence, the compositions of $X = (X_\mathcal{S} : k \in \mathcal{S})$ for different demand matrices in which $\mathbf{d}_k$ is the same, are equivalent from the viewpoint of user k.

In Appendix C, we will prove the privacy in a formal information-theoretic way.

Performance. For any demand matrix, we transmit $\binom{\mathsf{K}}{t+1} + \cdots + \binom{\mathsf{K}}{\mathsf{K}}$ multicast messages, each of which contains L linear combinations of subfiles. Since each subfile has $\frac{\mathsf{B}}{2^{\mathsf{K}-1}+\binom{\mathsf{K}-1}{t}+\cdots+\binom{\mathsf{K}-1}{\mathsf{K}-1}}$ bits, the achieved load is

$$\mathsf{L} \frac{\binom{\mathsf{K}}{t+1} + \cdots + \binom{\mathsf{K}}{\mathsf{K}}}{2^{\mathsf{K}-1} + \binom{\mathsf{K}-1}{t} + \cdots + \binom{\mathsf{K}-1}{\mathsf{K}-1}} = \mathsf{L} \frac{2^{\mathsf{K}} - \binom{\mathsf{K}}{0} - \cdots - \binom{\mathsf{K}}{t}}{2^{\mathsf{K}-1} + \binom{\mathsf{K}-1}{t} + \cdots + \binom{\mathsf{K}-1}{\mathsf{K}-1}},$$

as in (17). The sub-packetzation level is 2^{K}.

Remark 2. *It can be seen that in both of the above proposed schemes in Sections IV-A and IV-B, the placement precoding which leads that from the viewpoint of one user each cached subfile of one file is equivalent while each uncached subfile of one file is also equivalent, is the key to preserve the privacy of the demands of other users from this user. We refer this precoding as to Private Placement Precoding, which can be generalized as follows.*

We focus on a caching placement with a (n, k) MDS precoding where $n \geq k$. Each file F_i is divided into k non-overlapping and equal-length pieces, which are then encoded by a (n, k) MDS code. The MDS coded symbols of F_i is denoted by $S_1^i, \ldots, S_n^i$, each of which contains B/k bits. We randomly generate a permutation of $[n]$, denoted by $\mathbf{p}_i = (p_{i,1}, \ldots, p_{i,n})$, independently and uniformly over the set of all possible permutations. For each $j \in [n]$, we generate one subfile of each file F_i,

$$f_{i,\mathcal{W}_j} := S^i_{p_{i,j}}, \tag{44}$$

where $\mathcal{W}_j \subseteq [\mathsf{K}]$ and we let each user in $\mathcal{W}_j$ cache $f_{i,\mathcal{W}_j}$. As a result, from the viewpoint of user k, each cached subfile of F_i is equivalent from the viewpoint of user k, while each uncached subfile of F_i is also equivalent. It is obvious that when $n = k$, the placement is uncoded. Hence, the proposed private placement precoding can be also used with any uncoded cache placement.

Even if we use the proposed private precoding for the MAN coded caching scheme described in Section II-B, the privacy constraint does not hold because the compositions of the MAN multicast messages are not symmetric for different demand matrices. ☐

C. Proof of (18)

When $\mathsf{M} \geq \frac{2^{\mathsf{K}-1}}{2^{\mathsf{K}-1}+1}\mathsf{N}$, by memory-sharing between the corner points $t = \mathsf{K} - 1$ and $t = \mathsf{K}$ in (17), the MDS-based scheme in Section IV-C achieves the load $\mathsf{L}\left(1 - \frac{\mathsf{M}}{\mathsf{N}}\right)$, which coincides the converse bound for the MAN caching model with multiple request in [33].

In the following, we will introduce another private caching scheme for $\mathsf{M} = \frac{2\mathsf{K}-1}{2\mathsf{K}}\mathsf{N}$. By memory-sharing between the corner points $\mathsf{M} = \frac{2\mathsf{K}-1}{2\mathsf{K}}\mathsf{N}$ and $\mathsf{M}_2 = \mathsf{N}$, for any $\mathsf{M}_1 \geq \frac{2\mathsf{K}-1}{2\mathsf{K}}\mathsf{N}$, the load $\mathsf{L}\left(1 - \frac{\mathsf{M}_1}{\mathsf{N}}\right)$ is achievable. Hence, if $\frac{2\mathsf{K}-1}{2\mathsf{K}}\mathsf{N} \leq \frac{2^{\mathsf{K}-1}}{2^{\mathsf{K}-1}+1}\mathsf{N}$ (i.e., $\mathsf{K} \geq 4$), we can replace the corner point in (17) with $t = \mathsf{K} - 1$ by the corner point in (18).

Placement Phase. Each file F_i where $i \in [\mathsf{N}]$ is divided into $\binom{\mathsf{K}}{\mathsf{K}-1} + \mathsf{K}\binom{\mathsf{K}}{\mathsf{K}} = 2\mathsf{K}$ non-overlapping and equal-length pieces, denoted by $S_1^i, \ldots, S_{2\mathsf{K}}^i$, where each piece has $\frac{B}{2\mathsf{K}}$ bits. We randomly generate a permutation of $[2\mathsf{K}]$, denoted by $\mathbf{p}_i = (p_{i,1}, \ldots, p_{i,2\mathsf{K}})$, independently and uniformly over the set of all possible permutations. For each $k \in [\mathsf{K}]$, we generate one subfile $f_{i,[\mathsf{K}]\setminus\{k\}} = S_{p_{i,k}}^i$. In addition, for each $q \in [\mathsf{K}]$, we also generate one subfile $f_{i,[\mathsf{K}],q} = S_{p_{i,\mathsf{K}+q}}^i$.

Each user $k \in [\mathsf{K}]$ caches $f_{i,\mathcal{W}}$ where $\mathcal{W} \subseteq [\mathsf{K}]$ and $|\mathcal{W}| = \mathsf{K} - 1$, if $k \in \mathcal{W}$. User k also caches $f_{i,[\mathsf{K}],q}$ for each $q \in [\mathsf{K}]$. Hence, each user totally caches $\frac{\binom{\mathsf{K}-1}{\mathsf{K}-2} + \mathsf{K}\binom{\mathsf{K}-1}{\mathsf{K}-1}}{2\mathsf{K}}\mathsf{N}B = \mathsf{M}B$ bits in its cache, satisfying the memory size constraint.

Delivery Phase for $\mathbb{D}$. Notice that each user caches $2\mathsf{K} - 1$ subfiles of each file, and thus it needs to recover one subfile of each of its desired files.

In the delivery phase, only one multicast message is generated and transmitted by the server,

$$X = \mathbb{G}_{\mathsf{L}\times\mathsf{KN}}\ \mathbf{g}_{\mathbb{D}}. \tag{45}$$

$\mathbf{g}_{\mathbb{D}}$ is a vector containing KN pieces. Define $\mathbf{g}_{\mathbb{D}}(j)$ as the jth piece of $\mathbf{g}_{\mathbb{D}}$, where $j \in [\mathsf{KN}]$. For each $i \in [\mathsf{N}]$ and each $k \in [\mathsf{K}]$,

- if $i \in \mathbf{d}_k$ (i.e., user k demands F_i), we let $\mathbf{g}_{\mathbb{D}}\big((i-1)\mathsf{K} + k\big) = f_{i,[\mathsf{K}]\setminus\{k\}}$;
- otherwise, we let $\mathbf{g}_{\mathbb{D}}\big((i-1)\mathsf{K} + k\big) = f_{i,[\mathsf{K}],k}$.

Decodability. Among the KN subfiles in X, each user $k \in [\mathsf{K}]$ caches all except $f_{i,[\mathsf{K}]\setminus\{k\}}$ where $i \in \mathbf{d}_k$. By the property of $\mathbb{G}_{\mathsf{L}\times\mathsf{KN}}$ (each L columns are linearly independent), user k can recover these L subfiles. Hence, we prove the decodability.

Privacy. Let us focus on user k. Intuitively, for any demand matrix, there are exactly K subfiles of each file in $P_{[K]}$. Among the K subfiles of each file demanded by user k, user k caches $K-1$ subfiles, while among the K subfiles of each file not demanded by user k, user k caches K subfiles. In addition, for any $i \in [N]$, each subfile of F_i cached by user k is equivalent from the viewpoint of user k. Hence, the multicast message X for different demand matrices in which $\mathbf{d}_k$ is the same, are equivalent from the viewpoint of user k.

In Appendix D, we will prove the privacy in a formal information-theoretic way.

Performance. For any demand matrix, $P_{[K]}$ contains L linear combinations of subfiles. Since each subfile has $\frac{B}{2K}$ bits, the achieved load is $\frac{L}{2K}$, as in (18). The sub-packetzation level is $2K$.

V. CONCLUSIONS

In this paper, we introduced a novel shared-link caching model with private demands, while the objective is to design a two-phase caching scheme with minimum load while preserving the privacy of the users demands. We believe that preserving the privacy of the users demands from other users that legitimately use the caching/content delivery system is an important problem that differs conceptionally from previously proposed models with eavesdroppers or private information retrieval (PIR), as shortly outlined in Section I. For the formulated shared-link caching problem with private demands, we proposed two novel private coded caching schemes, the virtual-user scheme and the MDS-based scheme, which are information-theoretically private. Compared to the existing converse bounds for the shared-link caching model without privacy constraint, the virtual-user scheme is order optimal within a constant factor when $N \leq LK$, or when $N < LK$ and $M \geq N/K$. In addition, both of the two schemes are order optimal within a factor of 2 when $M \geq N/2$.

The only open case where the multiplicative gaps between the proposed schemes and the existing converse bounds for the shared-link caching model without privacy constraint are not constant is when $N < LK$ and $M < N/K$. In addition, since the virtual-user scheme has exponentially high sub-packetization level compared to the original MAN coded caching scheme and the MDS-based scheme does not have the order optimality results on the achieved load when $M < N/2$, the problem of preserving the privacy of the demands in the regime $M < N/2$ with order optimal load and small sub-packetization (at least not exponentially larger than the original MAN coded caching scheme remains open. On-going/future work includes deriving a converse

bound for this caching model with privacy and designing improved private caching schemes with small sub-packetization to solve the above two open problems.

APPENDIX A
PROOF OF ORDER OPTIMALITY RESULTS

A. Proof of Theorem 3

Converse. We use the existing converse bound in [4], [5], [8] for the shared-link caching model without privacy, which obviously provides a load lower bound for the shared-link caching model with private demands. More precisely, we consider $\mathsf{L} = 1$ and $\mathsf{L} > 1$, respectively.

- $\mathsf{L} = 1$. If $\mathsf{N} \le \mathsf{K}$, the lower convex envelope of $(0, \mathsf{N})$ and $\left(\frac{\mathsf{N}t'}{\mathsf{K}}, \frac{\mathsf{K}-t'}{t'+1}\right)$ where $t' \in [\mathsf{K}]$ is order optimal within a factor of 4 [5]. If $\mathsf{N} > \mathsf{K}$, the lower convex envelope of $\left(\frac{\mathsf{N}t'}{\mathsf{K}}, \frac{\mathsf{K}-t'}{t'+1}\right)$ where $t' \in [0 : \mathsf{K}]$ is order optimal within a factor of 2 [4] . In addition, as shown in [3] that the corner points $\left(\frac{\mathsf{N}t'}{\mathsf{K}}, \frac{\mathsf{K}-t'}{t'+1}\right)$ where $t' \in [0 : \mathsf{K}]$ are successively convex. Hence, when $\mathsf{N} > \mathsf{K}$ and $\mathsf{M} \ge \mathsf{N}/\mathsf{K}$, the lower convex envelop of $\left(\frac{\mathsf{N}t'}{\mathsf{K}}, \frac{\mathsf{K}-t'}{t'+1}\right)$, where $t' \in [\mathsf{K}]$ is order optimal within a factor of 2.

- $\mathsf{L} > 1$. The lower convex envelope of $(0, \mathsf{N})$ and $\left(\frac{\mathsf{N}t'}{\mathsf{K}}, \mathsf{L}\frac{\mathsf{K}-t'}{t'+1}\right)$ where $t' \in [0 : \mathsf{K}]$ is order optimal within a factor of 11 [8]. If $\mathsf{N} \le \mathsf{LK}$, the lower convex envelope of $(0, \mathsf{N})$ and $\left(\frac{\mathsf{N}t'}{\mathsf{K}}, \mathsf{L}\frac{\mathsf{K}-t'}{t'+1}\right)$ where $t' \in [\mathsf{K}]$ is order optimal within a factor of 11; if $\mathsf{N} > \mathsf{LK}$, the lower convex envelop of $\left(\frac{\mathsf{N}t'}{\mathsf{K}}, \mathsf{L}\frac{\mathsf{K}-t'}{t'+1}\right)$ where $t' \in [\mathsf{K}]$, is order optimal within a factor of 2 when $\mathsf{M} \ge \mathsf{N}/\mathsf{K}$.

Achievability. We will prove that from the achieved corner points by the proposed scheme in Theorem 2, $\left(\frac{\mathsf{N}t}{\binom{\mathsf{N}}{\mathsf{L}}\mathsf{K}}, \mathsf{L}\frac{\binom{\mathsf{N}}{\mathsf{L}}\mathsf{K}-t}{t+1}\right)$ where $t \in \left[\binom{\mathsf{N}}{\mathsf{L}}\mathsf{K}\right]$, we can achieve $\left(\frac{\mathsf{N}t'}{\mathsf{K}}, 2\mathsf{L}\frac{\mathsf{K}-t'}{t'+1}\right)$, where $t' \in [\mathsf{K}]$.

We now focus on one $t' \in [\mathsf{K}]$. We let $t = \binom{\mathsf{N}}{\mathsf{L}}t'$ and we can achieve

$$
\begin{aligned}
\mathsf{R}_{\mathsf{v}} &= \mathsf{L}\frac{\binom{\mathsf{N}}{\mathsf{L}}\mathsf{K} - t}{t + 1} \\
&= \mathsf{L}\frac{\binom{\mathsf{N}}{\mathsf{L}}\mathsf{K} - \binom{\mathsf{N}}{\mathsf{L}}t'}{\binom{\mathsf{N}}{\mathsf{L}}t' + 1} \\
&= \mathsf{L}\frac{\mathsf{K} - t'}{t' + \frac{1}{\binom{\mathsf{N}}{\mathsf{L}}}} \\
&\le \frac{2(\mathsf{K} - t')}{t' + 1},
\end{aligned}
\tag{46}
$$

where (46) comes from

$$\frac{t'+1}{t'+\frac{1}{\binom{N}{L}}} \leq \frac{t'+1}{t'} \leq 2, \text{ when } t \geq 1.$$

Recall that $(0, N)$ can be also achieved by the proposed scheme. Hence, we prove Theorem 3.

B. Proof of Theorem 5

Converse. We use the existing converse bound in [33] for the shared-link caching model without privacy, which obviously provides a load lower bound for the shared-link caching model with private demands. From [33, Theorem 1] with $s = 1$, we have

$$R^\star \geq L\left(1 - \frac{M}{N}\right). \tag{47}$$

Achievability. When $M_1 = N/2$, from (17) with $t = 0$ achieved by the improved scheme, we have

$$R_m = L\frac{2^K - 1}{2^K} \leq L. \tag{48}$$

Hence, by memory-sharing between $M_1 = N/2$ with load less than L and $M_2 = N$ with load equal to 0, we have for any $M \in [N/2, N]$,

$$R_m \leq 2L\left(1 - \frac{M}{N}\right)$$

$$\leq 2R^\star, \tag{49}$$

where (49) comes from (47).

Similarly, by letting $t = \left\lfloor \frac{\binom{N}{L}K}{2} \right\rfloor$ in (16), it can be proved that when $M_1 = N/2$, $R_v \leq L$. Hence, R_v is also order optimal within a factor of 2 when $M \geq N/2$.

APPENDIX B

PROOF OF LEMMA 1

It is equivalent to prove for any two sets $\mathcal{S}_1 \subseteq [K]$ and $\mathcal{S}_2 \subseteq [K]$ where $\mathcal{S}_1 \neq \mathcal{S}_2$, we have

$$(\mathcal{S}_1 \cup \mathcal{Q}_i) \setminus (\mathcal{S}_1 \cap \mathcal{Q}_i) \neq (\mathcal{S}_2 \cup \mathcal{Q}_i) \setminus (\mathcal{S}_2 \cap \mathcal{Q}_i), \ \forall i \in [N]. \tag{50}$$

In addition, for each $j \in [2]$, we let $\mathcal{S}_j = \mathcal{S}_{j,1} \cup \mathcal{S}_{j,2}$ where $\mathcal{S}_{j,1} \subseteq \mathcal{Q}_i$ and $\mathcal{S}_{j,2} \cap \mathcal{Q}_i = \emptyset$.

Without loss of generality, we assume $|\mathcal{S}_1| \geq |\mathcal{S}_2|$. We focus on two cases:

1) $\mathcal{S}_{1,2} \neq \mathcal{S}_{2,2}$. It can be seen that $\mathcal{S}_{j,2} \subseteq ((\mathcal{S}_j \cup \mathcal{Q}_i) \setminus (\mathcal{S}_j \cap \mathcal{Q}_i))$ for each $j \in [2]$. Hence, (50) holds for this case.

2) $\mathcal{S}_{1,2} = \mathcal{S}_{2,2}$ and $\mathcal{S}_{1,1} \neq \mathcal{S}_{2,1}$. Since $|\mathcal{S}_1| \geq |\mathcal{S}_2|$ and $\mathcal{S}_{1,1} \neq \mathcal{S}_{2,1}$, there exists at least one user in $\mathcal{Q}_i$ (assume to be k) who is in $\mathcal{S}_{1,1} \setminus \mathcal{S}_{2,1}$. Hence, this user k is in $(\mathcal{S}_2 \cup \mathcal{Q}_i) \setminus (\mathcal{S}_2 \cap \mathcal{Q}_i)$ but not in $(\mathcal{S}_1 \cup \mathcal{Q}_i) \setminus (\mathcal{S}_1 \cap \mathcal{Q}_i)$. Hence, (50) holds for this case.

In conclusion, we prove Lemma 1.

APPENDIX C

PROOF OF THE PRIVACY FOR THE NEW SCHEME IN (17)

We consider $t = 0$ in (17). It can be seen when $t > 0$, the transmitted multicast messages are included in the the transmitted multicast messages for $t = 0$. Hence, if we prove the privacy for $t = 0$, the privacy for $t > 0$ can also be proved.

For the new scheme in (17), we want to prove the privacy constraint in (11),

$$I(\mathbb{D}_{\setminus \{k\}}; X | Z_k, \mathbf{d}_k) = 0, \ \forall k \in [\mathsf{K}]. \tag{51}$$

We now focus on one user k, one demand vector $\mathbf{d}_k$, and one cache realization z_k. Assume $(x_{\mathcal{S}} : \mathcal{S} \subseteq [\mathsf{K}], |\mathcal{S}| > 0)$ is a possible realization of $(X_{\mathcal{S}} : \mathcal{S} \subseteq [\mathsf{K}], |\mathcal{S}| > 0)$, given $\mathbf{d}_k$ and z_k. We want to prove for any demand matrix $\mathbb{D}_{\setminus \{k\}}$, the probability

$$\Pr\{(X_{\mathcal{S}} : \mathcal{S} \subseteq [\mathsf{K}], |\mathcal{S}| > 0) = (x_{\mathcal{S}} : \mathcal{S} \subseteq [\mathsf{K}], |\mathcal{S}| > 0) | \mathbf{d}_k, z_k, \mathbb{D}_{\setminus \{k\}}\}$$

does not depend on $\mathbb{D}_{\setminus \{k\}}$.

For each $\mathcal{S} \subseteq [\mathsf{K}]$ and each $i \in [\mathsf{N}]$, we denote the coded MDS symbol of F_i in $X_{\mathcal{S}}$ by $X_{\mathcal{S},i}$. We have

$$\Pr\{(X_{\mathcal{S}} : \mathcal{S} \subseteq [\mathsf{K}], |\mathcal{S}| > 0) = (x_{\mathcal{S}} : \mathcal{S} \subseteq [\mathsf{K}], |\mathcal{S}| > 0) | \mathbf{d}_k, z_k, \mathbb{D}_{\setminus \{k\}}\}$$

$$= \Pr\{(X_{\mathcal{S},i} : \mathcal{S} \subseteq [\mathsf{K}], |\mathcal{S}| > 0, i \in [\mathsf{N}]) = (x_{\mathcal{S},i} : \mathcal{S} \subseteq [\mathsf{K}], |\mathcal{S}| > 0, i \in [\mathsf{N}]) | \mathbf{d}_k, z_k, \mathbb{D}_{\setminus \{k\}}\}$$

$$= \prod_{i \in [\mathsf{N}]} \Pr\{(X_{\mathcal{S},i} : \mathcal{S} \subseteq [\mathsf{K}], |\mathcal{S}| > 0) = (x_{\mathcal{S},i} : \mathcal{S} \subseteq [\mathsf{K}], |\mathcal{S}| > 0) | \mathbf{d}_k, z_k, \mathbb{D}_{\setminus \{k\}}\}, \tag{52}$$

where (52) comes from that the placement permutations $\mathbf{p}_1, \ldots, \mathbf{p}_\mathsf{N}$ are independent.

We then focus on two cases:

- $i \in \mathbf{d}_k$. It is claimed in Lemma 1 that there does not exist any subfile appearing two multicast messages. Given z_k, there are $2^{\mathsf{K}-1}$ MDS coded symbols of F_i not in z_k, each of which should be in one different $X_{\mathcal{S}_1}$ where $\mathcal{S}_1 \subseteq [\mathsf{K}]$ and $k \in \mathcal{S}_1$. In addition, there are

2^{K-1} MDS coded symbols of F_i in z_k, each of which should be in one different $X_{\mathcal{S}_2}$ where $\mathcal{S}_2 \subseteq [K]$, $k \notin \mathcal{S}_2$, and $|\mathcal{S}_2| > 0$. Hence, we have

$$\Pr\left\{(X_{\mathcal{S},i} : \mathcal{S} \subseteq [K], |\mathcal{S}| > 0) = (x_{\mathcal{S},i} : \mathcal{S} \subseteq [K], |\mathcal{S}| > 0)\Big|\mathbf{d}_k, z_k, \mathbb{D}_{\backslash\{k\}}\right\} \tag{53a}$$

$$= \Pr\left\{(X_{\mathcal{S}_1,i} : \mathcal{S}_1 \subseteq [K], k \in \mathcal{S}_1) = (x_{\mathcal{S}_1,i} : \mathcal{S}_1 \subseteq [K], k \in \mathcal{S}_1),\right.$$
$$\left.(X_{\mathcal{S}_2,i} : \mathcal{S}_2 \subseteq [K], k \notin \mathcal{S}_2, |\mathcal{S}_2| > 0) = (x_{\mathcal{S}_2,i} : \mathcal{S}_2 \subseteq [K], k \notin \mathcal{S}_2, |\mathcal{S}_2| > 0)\Big|\mathbf{d}_k, z_k, \mathbb{D}_{\backslash\{k\}}\right\} \tag{53b}$$

$$= \Pr\left\{(X_{\mathcal{S}_1,i} : \mathcal{S}_1 \subseteq [K], k \in \mathcal{S}_1) = (x_{\mathcal{S}_1,i} : \mathcal{S}_1 \subseteq [K], k \in \mathcal{S}_1)\Big|\mathbf{d}_k, z_k, \mathbb{D}_{\backslash\{k\}}\right\}$$
$$\Pr\left\{(X_{\mathcal{S}_2,i} : \mathcal{S}_2 \subseteq [K], k \notin \mathcal{S}_2, |\mathcal{S}_2| > 0) = (x_{\mathcal{S}_2,i} : \mathcal{S}_2 \subseteq [K], k \notin \mathcal{S}_2, |\mathcal{S}_2| > 0)\Big|\mathbf{d}_k, z_k, \mathbb{D}_{\backslash\{k\}}\right\} \tag{53c}$$

$$= \left(\frac{1}{2^{K-1}!}\right)^2, \tag{53d}$$

where ! represents the factorial operation. (53c) comes from that each $X_{\mathcal{S}_1,i}$ is not cached by user k and each $X_{\mathcal{S}_2,i}$ is cached by user k, and thus their realizations are independent given z_k and $\mathbb{D}$.

- $i \notin \mathbf{d}_k$. It is claimed in Lemma 1 that there does not exist any subfile appearing two multicast messages. Given z_k, there are 2^{K-1} coded MDS symbols of F_i in z_k, each of which should be in one different $X_{\mathcal{S}_1}$ where $\mathcal{S} \subseteq [K]$ and $k \in \mathcal{S}_1$. In addition, there are 2^{K-1} coded MDS symbols of F_i not in z_k, each of which should be in one different $X_{\mathcal{S}_2}$ where $\mathcal{S} \subseteq [K]$, $k \notin \mathcal{S}_2$, and $|\mathcal{S}_2| > 0$. Hence, from the same derivation as (53d) we have

$$\Pr\{(X_{\mathcal{S},i} : \mathcal{S} \subseteq [K], |\mathcal{S}| > 0) = (x_{\mathcal{S},i} : \mathcal{S} \subseteq [K], |\mathcal{S}| > 0)|\mathbf{d}_k, z_k, \mathbb{D}_{\backslash\{k\}}\} = \left(\frac{1}{2^{K-1}!}\right)^2. \tag{54}$$

It can be seen both of the probabilities in (53d) and (54) are independent of $\mathbb{D}_{\backslash\{k\}}$. Hence, we can prove the probability in (52) is also independent of $\mathbb{D}_{\backslash\{k\}}$. In conclusion, we prove the privacy constraint in (11).

APPENDIX D

PROOF OF THE PRIVACY FOR THE NEW SCHEME IN (18)

For the new scheme in (18), there is only one multicast message in X. We want to prove the privacy constraint in (11),

$$I(\mathbb{D}_{\backslash\{k\}}; X|Z_k, \mathbf{d}_k) = 0, \ \forall k \in [K]. \tag{55}$$

We also focus on one user k, one demand vector $\mathbf{d}_k$, and one cache realization z_k. Assume x is a possible realization of X, given $\mathbf{d}_k$ and z_k. We want to prove for any demand matrix $\mathbb{D}_{\backslash\{k\}}$, the probability

$$\Pr\{X = x | \mathbf{d}_k, z_k, \mathbb{D}_{\backslash\{k\}}\}$$

does not depend on $\mathbb{D}_{\backslash\{k\}}$.

In X, there are K pieces of each file F_i. Hence, X_i now denotes the set of K pieces of F_i in X. Since the placement permutations $\mathbf{p}_1, \ldots, \mathbf{p}_\mathsf{N}$ are independent, we have

$$\Pr\{X = x | \mathbf{d}_k, z_k, \mathbb{D}_{\backslash\{k\}}\} = \prod_{i \in [\mathsf{N}]} \Pr\{X_i = x_i | \mathbf{d}_k, z_k, \mathbb{D}_{\backslash\{k\}}\}. \tag{56}$$

We also focus two cases:

- $i \in \mathbf{d}_k$. Notice that z_k contains $2\mathsf{K} - 1$ pieces of F_i while F_i contains $2\mathsf{K}$ pieces. In addition, in X_i there are $\mathsf{K} - 1$ pieces of F_i cached in z_k and one piece of F_i not cached in z_k. Hence, we have

$$\Pr\{X_i = x_i | \mathbf{d}_k, z_k, \mathbb{D}_{\backslash\{k\}}\} = \frac{1}{\binom{2\mathsf{K}-1}{\mathsf{K}-1}}. \tag{57}$$

- $i \notin \mathbf{d}_k$. In X_i there are K pieces of F_i cached in z_k. Hence, we have

$$\Pr\{X_i = x_i | \mathbf{d}_k, z_k, \mathbb{D}_{\backslash\{k\}}\} = \frac{1}{\binom{2\mathsf{K}-1}{\mathsf{K}}} = \frac{1}{\binom{2\mathsf{K}-1}{\mathsf{K}-1}}. \tag{58}$$

It can be seen both of the probabilities in (57) and (58) are independent of $\mathbb{D}_{\backslash\{k\}}$. Hence, we can prove the probability in (56) is also independent of $\mathbb{D}_{\backslash\{k\}}$. In conclusion, we prove the privacy constraint in (11).

REFERENCES

[1] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5g wireless networks," *IEEE Communications Magazine*, vol. 52, pp. 82–89, Aug. 2014.

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

[3] K. Wan, D. Tuninetti, and P. Piantanida, "On the optimality of uncoded cache placement," *in IEEE Infor. Theory Workshop*, Sep. 2016.

[4] Q. Yu, M. A. Maddah-Ali, and S. Avestimehr, "Characterizing the rate-memory tradeoff in cache networks within a factor of 2," *in IEEE Int. Symp. Inf. Theory*, Jun. 2017.

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

[6] Q. Yu, M. A. Maddah-Ali, and S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *IEEE Trans. Infor. Theory*, vol. 64, pp. 1281 – 1296, Feb. 2018.

[7] M. Ji, A. Tulino, J. Llorca, and G. Caire, "Caching-aided coded multicasting with multiple random requests," *in Proc. IEEE Inf. Theory Workshop (ITW)*, May. 2015.

[8] A. Sengupta and R. Tandon, "Improved approximation of storage-rate tradeoff for caching with multiple demands," *IEEE Trans. Commun.*, vol. 65, no. 5, pp. 1940–1955, May. 2017.

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

[10] M. Ji, G. Caire, and A. Molisch, "Fundamental limits of caching in wireless d2d networks," *IEEE Trans. Inf. Theory*, vol. 62, no. 1, pp. 849–869, 2016.

[11] S. P. Shariatpanahi, S. A. Motahari, and B. H. Khalaj, "Multi-server coded caching," *IEEE Trans. Infor. Theory*, vol. 62, pp. 7253 – 7271, Dec. 2016.

[12] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Caching in combination networks," *49th Asilomar Conf. on Sig., Sys. and Comp.,*, Nov. 2015.

[13] K. Wan, M. Ji, P. Piantanida, and D. Tuninetti, "Caching in combination networks: Novel multicast message generation and delivery by leveraging the network topology," *in IEEE Intern. Conf. Commun (ICC 2018)*, May 2018.

[14] A. Sengupta, R. Tandon, and T. C. Clancy, "Fundamental limits of caching with secure delivery," *IEEE Trans. on Information Forensics and Security*, vol. 10, no. 2, pp. 355–370, 2015.

[15] M. Bahrami, M. A. Attia, R. Tandon, and B. Vasic, "Towards the exact rate-memory trade-off for uncoded caching with secure delivery," *in 55th Annual Allerton Conf. on Commun., Control, and Computing (Allerton)*, Oct. 2017.

[16] V. Ravindrakumar, P. Panda, N. Karamchandani, and V. M. Prabhakaran, "Private coded caching," *IEEE Trans. on Information Forensics and Security*, vol. 13, no. 3, pp. 685–694, 2018.

[17] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[18] A. A. Zewail and A. Yener, "Device-to-device secure coded caching," *arXiv:1809.06844*, Sep. 2018.

[19] Z. H. A. . R. Mathar, "Bounds on caching d2d networks with secure delivery," *in 15th Int. Symp. Wireless Commun. Sys. (ISWCS)*, Aug. 2018.

[20] A. A. Zewail and A. Yener, "Combination networks with or without secrecy constraints: The impact of caching relays," *in IEEE Journal on Selected Areas in Communications*, vol. 36, no. 6, pp. 1140–1152, 2018.

[21] S. Kamel, M. Sarkiss, M. Wigger, and G. R. Othman, "Secrecy capacity-memory tradeoff of erasure broadcast channels," *IEEE Trans. Inf. Theory*, vol. 65, no. 8, pp. 5094–5124, 2019.

[22] F. Engelmann and P. Elia, "A content-delivery protocol, exploiting the privacy benefits of coded caching," *2017 15th Intern. Symp. on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, May 2017.

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

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

[25] Z. Chen, Z. Wang, and S. Jafar, "The capacity of private information retrieval with private side information," *available at arXiv:1709.03022*, Sep. 2017.

[26] S. Li and M. Gastpar, "Single-server multi-user private information retrieval with side information," *in IEEE Int. Symp. Inf. Theory*, Jun. 2018.

[27] R. Tandon, "The capacity of cache aided private information retrieval," *in 55th Allerton Conf. Commun., Control, Comp.*, Oct. 2017.

[28] Y.-P. Wei, K. Banawan, and S. Ulukus, "Cache-aided private information retrieval with partially known uncoded prefetching: Fundamental limits," *available at arXiv:1712.07021*, Dec. 2017.

[29] ——, "Fundamental limits of cache-aided private information retrieval with unknown and uncoded prefetching," *available at arXiv:1709.01056*, Sep. 2017.

[30] M. A. Attia, D. Kumar, and R. Tandon, "The capacity of private information retrieval from uncoded storage constrained databases," *available at arXiv:1805.04104*, May 2018.

[31] H. Sun and S. A. Jafar, "The capacity of private computation," *IEEE Trans. Inf. Theory*, vol. 65, no. 5, pp. 3880–3897, Jun. 2019.

[32] A. E. Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge, UK: Cambridge University Press, 2011.

[33] K. Wan, D. Tuninetti, M. Ji, and G. Caire, "Novel inter-file coded placement and d2d delivery for a cache-aided fog-ran architecture," *arXiv:1811.05498*, Nov. 2018.

[34] K. Wan, D. Tuninetti, and P. Piantanida, "On caching with more users than files," *in IEEE Int. Symp. Inf. Theory*, Jul. 2016.