

Subpacketization in Coded Caching with Demand Privacy

Aravind V R* Pradeep Sarvepalli† Andrew Thangaraj‡

Department of Electrical Engineering, Indian Institute of Technology Madras, India

Email: *ee13d205@ee.iitm.ac.in, †pradeep@ee.iitm.ac.in, ‡andrew@ee.iitm.ac.in

Abstract—Coded caching is a technique where we utilize multi-casting opportunities to reduce rate in cached networks. One limitation of coded caching schemes is that they reveal the demands of all users to their peers. In this work, we consider coded caching schemes that assure privacy for user demands with a particular focus on reducing subpacketization. For the 2-user, 2-file case, we present a new linear demand-private scheme with the lowest possible subpacketization. This is done by presenting the scheme explicitly and proving impossibility results under lower subpacketization. Additionally, when only partial privacy is required, we show that subpacketization can be significantly reduced when there are a large number of files.

I. INTRODUCTION

Data traffic has been growing rapidly in recent years with content delivery, especially that of multimedia files, contributing a significant part. One important aspect of such traffic is its temporal variation. Network usage during peak demand times could be much higher than the demand in off-peak hours. Caching is a way to alleviate network congestion during peak hours by prefetching popular content nearer to the user during off-peak hours. Depending on the limitations on memory, a part of these files would be prefetched and once the user makes a demand, the rest of the requested file will be transmitted. Early literature on caching focused on cache placement/replacement policies [1], caching architectures [3], [14], [16], web request models [2] etc.

Maddah-Ali and Niesen had shown in their seminal paper that coding can achieve significant gain over uncoded caching by making use of multicast opportunities [12]. Coded caching achieves an additional *global caching gain*, which is proportional to the number of users. Their scheme is shown to be order optimal with an information-theoretic lower bound on the number of files needed to be transmitted (known as *rate*). Though the exact lower bound on peak rate is still an open problem several works had investigated this and came up with tighter bounds [7], [19], [26], [29]. The problem

has been studied in several settings like decentralized caching [13], non-uniform demands [15], multiple levels of cache [10] to name a few. Most of the schemes in these works involve storing the prefetched parts of files in uncoded form. Coded prefetching is investigated in [4], [8], [24], where linear combinations of subfiles are stored in caches. In a few regimes this approach can improve the rate-memory trade-off over uncoded prefetching.

Yan *et al.* developed a structure called *placement delivery arrays* that could model both the placement and delivery schemes in a single array [27]. Graphical models for caching have been investigated in [20], [22], [28]. Schemes can also be derived using combinatorial designs and linear block codes [23]. A limitation with the original centralized scheme was the high subpacketization of files [21]. In the original scheme due to [12], the number of subfiles a file is split into, increases exponentially with the number of users. These combinatorial models have helped in developing schemes that have lower subpacketization but with a small penalty on rate [5], [22].

One area of particular interest is security and privacy in coded caching. In typical coded caching schemes, other users involved in the multicast or eavesdroppers might get to know the identity of the file a particular user demanded and its contents. Furthermore, users will be able to partially access files which they have not demanded. This is in part due to the cache that contains contents of files not requested by them and also because, during delivery, they may be able to decode packets not meant for them. Sengupta *et al.* [18] proposed a method for preventing information leakage to an external wiretapper with the use of cryptographic keys. Visakh *et al.* [17] had recently shown that the contents of a file could be revealed only to the user/users who requested it, using secret sharing techniques.

One aspect that has not been investigated much is the privacy of the user requests in the specific context of coded caching, while it has been studied in

closely related areas like index coding [11] and private information retrieval (PIR) [6]. As we were preparing this manuscript, we became aware of work due to Wan and Caire [25] who take a different approach for user request privacy from ours. Another paper by Kamath [9] also addressed the problem of demand privacy and their approach is similar to the one in this work. We point out the specific differences in our results when compared to those from [25] and [9] below.

In this work, we explore methods to obtain privacy of each user's requests from the other users in coded caching keeping subpacketization constraints as an important parameter.

Our specific contributions are as follows:

- i) We focus on the 2-user, 2-file case in detail and provide an achievable multicast transmission rate versus cache storage curve under a demand privacy constraint.
- ii) For the 2-user, 2-file case with cache storage of 1 file, we show an explicit demand-private scheme achieving a multicast transmission rate of $2/3$ with a subpacketization of 3. This scheme cannot be obtained using the general scheme proposed in [9], which, in fact, requires a subpacketization of 6.
- iii) For the 2-user, 2-file case, we prove some impossibility results on subpacketization of 2 and uncoded cache storage for linear coded caching with demand privacy. These are some of the first negative results in this new area.
- iv) Finally, we propose a general K -user, N -file partially demand-private scheme that provides a trade-off between the level of privacy and reduction in subpacketization.

The rest of the paper is organized as follows. In Section II, we describe the system setup and the problem statement. In Section III, we provide demand-private schemes and an achievable rate vs cache memory curve for the case of two users and two files. We prove certain impossibility results with respect to packetization and coded prefetching. In Section IV, we describe the general scheme for constructing demand-private coded caching schemes from non-private coded caching schemes from [9], and provide specific instances of the construction from PDAs resulting in lesser subpacketization. We also introduce the notion of partially private schemes and show how to construct a partially private scheme. We conclude with a brief discussion on scope for future work in Section V.

II. PROBLEM STATEMENT

A. System setup

Assume that we have a server with N files. Each file is assumed to be of F bits and the i -th file is denoted W_i . The server is connected to K users via a multicast link. Each user has a cache of size MF bits. The cache contents of the i -th user are denoted Z_i . The system setup is shown in Fig. 1. The cache system works in

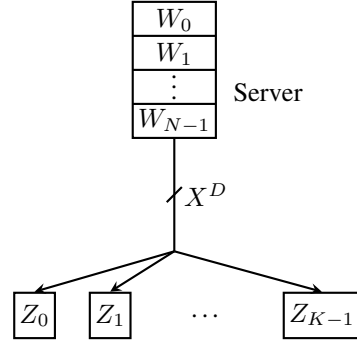


Fig. 1. Caching system.

two phases. In the first phase called the *placement phase*, the cache of each user is populated with content by the server. In addition, the server sends metadata or header information $\Theta(Z_i)$ about how the cache content was derived from the files to User i . The header information is assumed to be small in size when compared to the file size but crucial for decoding purposes. Note that during the placement phase the server is unaware of the files demanded by the users. We assume that the transmission of cache content and header takes place over a private link between the server and each user.

In the second phase, called the *delivery phase*, each user requests the server for one of the files from the set of N files. The demand of the i -th user is denoted D_i , where $D_i \in [N] \triangleq \{0, 1, \dots, N-1\}$. The demands of all the users 0 to $K-1$ is denoted by the demand vector $D = (D_0, D_1, \dots, D_{K-1})$. We assume that the D_i are all i.i.d. random variables uniformly distributed over $[N]$ and that the demands are sent over a private link between the user and the server. Based on the demands, the server multicasts ℓ packets, typically of the same size. The entire multicast transmission from the server is denoted X^D for a demand vector D . It consists of RF bits. The transmission X^D depends on the cache Z_i and the demands D_i . The quantity R is called the rate of transmission. In addition to X^D , some additional metadata or header information about the transmission is typically multicast in coded caching schemes. This metadata, denoted $\Theta(X^D)$, is usually small compared

to the file size and provides critical information for decoding by the users.

The main requirement in a coded caching scheme is that User i should be able to decode the file W_{D_i} using $Z_i, \Theta(Z_i), X^D$ and $\Theta(X^D)$. In other words, we require

$$H(W_{D_i} | Z_i, \Theta(Z_i), X^D, \Theta(X^D)) = 0. \quad (1)$$

We denote a coded caching scheme with K users, N files, local cache size M , and rate R as a $(K, N; M, R)$ coded caching scheme, or as a (K, N) scheme in short.

B. Demand privacy in coded caching

We will introduce the notion of demand privacy in coded caching with a simple example. Consider the $(2, 2)$ coded caching scheme due to Maddah-Ali and Niesen [12] shown in Fig. 2.

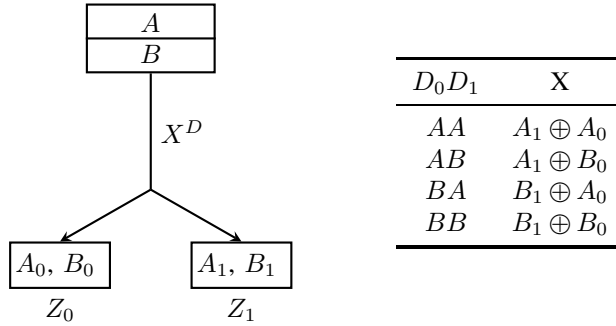


Fig. 2. Non-private scheme from [12] for $N = 2$ files, $K = 2$ users and demand vector, $D = (D_1, D_2)$.

Suppose that the demand is (A, A) . This results in the transmission $A_1 \oplus A_0$. To recover the files, each user must know what linear combination of subfiles has been transmitted. So, we will suppose that the server sends the linear combination information as header along with the transmission. It is easy to see that each user can recover the missing portion of the file demanded by them. However, the scheme has the unfortunate side effect of revealing the demands of each user to the other parties. From the header and scheme details, it is clear to User 0 that User 1 demanded the file A and vice versa.

If the transmission is $A_i \oplus B_j$, then the i -th user can infer that the j -th user has requested A based on the linear combination header information. In general, users can use the combined information of their cache, demands and header data from the server to learn about another user's demands.

Based on the preceding discussion, to achieve demand privacy in a coded caching scheme, we impose the following additional condition for all demand vectors D :

$$I(D_i, Z_i, \Theta(Z_i), X^D, \Theta(X^D); D_j) = 0, \quad i \neq j. \quad (2)$$

In other words, we require that the i -th user is completely uncertain about what the j -th user demands, given all information available to User i in the coded caching scheme. It can be shown that the standard Maddah-Ali-Niesen scheme [12] does not satisfy the demand privacy condition in Eq. (2).

III. $(K = 2, N = 2)$ CODED CACHING WITH DEMAND PRIVACY

We will first consider the case when there are two files and two users. A complete characterization of the M vs R region in the case of two files/users was one of the starting points of the area of coded caching. Therefore, it is important to fully characterize the same region with demand privacy. We have made some partial progress towards this problem.

First, we will show the design of a linear $(2, 2; 1, 2/3)$ coded caching scheme with demand privacy having a subpacketization (number of parts into which each file is divided) of 3. In comparison, directly converting a $(4, 2)$ -Maddah-Ali-Niesen scheme into a $(2, 2; 1, 2/3)$ demand-private scheme requires a subpacketization of 6 [9].

A. $(M = 1, R = 2/3)$ scheme with subpacketization 3

The two files, A and B , are divided into 3 parts A_i , $i = 0, 1, 2$ and B_i , $i = 0, 1, 2$. Table I summarizes the entire scheme.

TABLE I
 $(K = 2, N = 2; M = 1, R = 2/3)$ DEMAND-PRIVATE CACHING SCHEME WITH SUBPACKETIZATION 3. IF SERVER ASSIGNS THE CACHE Z_{i0} TO USER i , THEN $X^{D_0 D_1}$ IS THE TRANSMISSION FOR THE DEMAND $D_0 D_1$.

Notation	Possible cache contents	$D_0 D_1$	$X^{D_0 D_1}$
Z_{00}	$A_0 \oplus A_1$	AA	A_0
	$B_0 \oplus B_1$		
	$A_2 \oplus B_1$		
Z_{01}	$A_0 \oplus A_1$	AB	A_1
	$B_0 \oplus B_1$		
	$A_1 \oplus B_2$		
Z_{10}	$A_0 \oplus A_2$	BA	A_2
	$B_0 \oplus B_2$		
	$A_1 \oplus B_2$		
Z_{11}	$A_0 \oplus A_2$	BB	$A_0 \oplus A_1 \oplus A_2$
	$B_0 \oplus B_2$		
	$A_2 \oplus B_1$		

In the placement phase, the server places either Z_{i0} or Z_{i1} , with equal probability, as the cache Z_i for User i . The actual choice is private between the server and User i . If User i was assigned the cache Z_{i0} , then

TABLE II
FILES RECOVERED FROM POSSIBLE CACHE PAIRS AND
TRANSMISSION X FOR THE SCHEME FROM TABLE I

X Caches Z_0, Z_1	A_0 B_0	A_1 B_1	A_2 B_2	$A_0 \oplus A_1 \oplus A_2$ $B_0 \oplus B_1 \oplus B_2$
Z_{00}, Z_{10}	A, A	A, B	B, A	B, B
Z_{00}, Z_{11}	A, B	A, A	B, B	B, A
Z_{01}, Z_{11}	B, B	B, A	A, B	A, A
Z_{01}, Z_{10}	B, A	B, B	A, A	A, B

the multicast transmissions $X^{D_0 D_1}$ for each possible demand (D_0, D_1) are as shown in Table I. It can be seen that all the demands are served. It can also be checked that the demands are private under this assignment. For instance, from Table II, we see that there exists another assignment of cache for each user which recovers another file with the same transmission.

Table III is the set of recoverable files under each possible cache content for a given transmission. For the

TABLE III
FILES RECOVERED FROM POSSIBLE CACHES FOR A $(2, 2; 1, 2/3)$
PRIVATE SCHEME.

	X^{AB}	X^{BA}	X^{BB}	X^{AA}
Z_{00}	A	B	B	A
Z_{01}	B	A	A	B
Z_{10}	B	A	B	A
Z_{11}	A	B	A	B

same transmission, each user is able to recover either file A or file B with the two possible cache contents. Since the actual cache content is private, we readily see that this scheme satisfies the demand privacy condition in Eq. (2).

B. Dual private schemes

We show that a $(2, 2; M = M_1, R = R_1)$ scheme with demand privacy can be converted into a $(2, 2; M = R_1, R = M_1)$ demand-private scheme and this results in symmetric R vs M capacity bounds for the $(2, 2)$ case.

One can observe that the roles of caches and transmissions can be interchanged in the symmetric file recovery matrix in Table III. Hence, from the scheme given in Table I, we can arrive at a scheme given in Table IV with rate $R = 1$ for $M = 2/3$. We call this scheme the dual of the original scheme.

Our next result generalizes the above for all $(2, 2)$ private schemes that use one of two caches uniformly at random.

Lemma 1 (Duality of transmissions and caches). *Suppose that there exists a $(2, 2; M = M_1, R = R_1)$ private scheme where the server places one of two possible*

TABLE IV
DUAL PRIVATE $(2, 2; 2/3, 1)$ SCHEME FROM THE PRIVATE
 $(2, 2; 1, 2/3)$ SCHEME GIVEN IN TABLE I. FOR CACHE Z_{i0} AT
USER i , $X^{D_0 D_1}$ IS THE TRANSMISSION FOR THE DEMAND $D_0 D_1$.

Notation	Possible Contents	Cache	$D_1 D_2$	$X^{D_0 D_1}$
Z_{00}	A_1 B_1		AA	$A_0 \oplus A_2$ $B_0 \oplus B_2$ $A_2 \oplus B_1$
Z_{01}	A_2 B_2		AB	$A_0 \oplus A_1$ $B_0 \oplus B_1$ $A_2 \oplus B_1$
Z_{10}	$A_0 \oplus A_1 \oplus A_2$ $B_0 \oplus B_1 \oplus B_2$		BA	$A_0 \oplus A_1$ $B_0 \oplus B_1$ $A_1 \oplus B_2$
Z_{11}	A_0 B_0		BB	$A_0 \oplus A_2$ $B_0 \oplus B_2$ $A_1 \oplus B_2$

cache contents uniformly at random. Then, there exists a $(2, 2; M = R_1, R = M_1)$ private scheme.

Proof. Consider a $(2, 2; M, R)$ private scheme constructed with users having two options to populate their caches. Let $\{Z_{00}, Z_{01}\}$ be the set of two cache options for User 0 and $\{Z_{10}, Z_{11}\}$ be the set of two cache options for User 1. Let $X^{D_1 D_2}$ be the transmission corresponding to the user demands $D = (D_1, D_2)$ and cache Z_{i0} at User i . The sets $\mathcal{Z} = \{Z_{00}, Z_{01}, Z_{10}, Z_{11}\}$ and $\mathcal{X} = \{X^{AA}, X^{AB}, X^{BA}, X^{BB}\}$ are able to recover files A and B as given in Table III. Let the size of Z_i be $R_1 F$ bits and that of $X^{W_0 W_1}$ be $M_1 F$ bits. We can interchange the role of these caches and transmissions. Let $\{X^{AB}, X^{BA}\}$ be the set of two cache options for User 0 and $\{X^{BB}, X^{AA}\}$ be the set of two cache options for User 1. Then if Z_{11} is transmitted and the Users 0 and 1 are assigned $\{X^{AB}\}$ and $\{X^{BB}\}$ as their caches, both can recover file A . Instead if User 1 had X^{BA} in its cache, the users would have recovered B and A , respectively. This way of interchangeability between caches and transmissions gives rise to a new scheme for 2 users and 2 files, where the cache size is M_1 bits and transmission size is R_1 bits. \square

A consequence of the above duality is that the achievable trade-off between memory and rate for $(2, 2)$ private schemes is symmetric about the line $M = R$.

Lemma 2 (Time sharing with file splitting). *Given two achievable (M, R) pairs for a $(2, 2)$ private scheme, all values of (M, R) along the line joining these points are achievable.*

Proof. Consider $0 \leq \alpha \leq 1$. Split the file A into two parts A_α and $A_{\bar{\alpha}}$ of size αF bits and $(1 - \alpha)F$

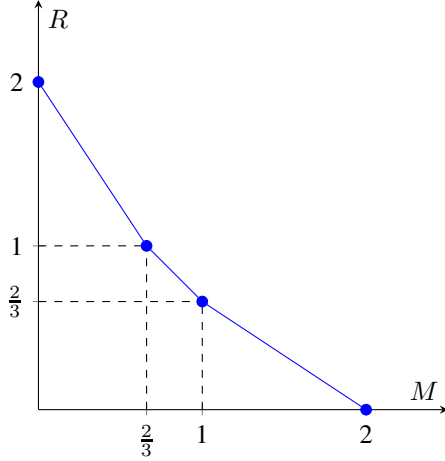


Fig. 3. Achievable (M, R) region for $(2, 2; M, R)$ private schemes. The $(2, 2; 1, 2/3)$ scheme and its dual scheme $(2, 2; 2/3, 1)$ have a subpacketization of three subfiles. The straight lines are due to Lemma 2. For these schemes, the subpacketization need not be three.

bits respectively. Similarly, split B into B_α and $B_{\bar{\alpha}}$. Denote the two achievable private caching schemes as $(2, 2; M, R)$ and $(2, 2; M', R')$ respectively. We can use the $(2, 2; M, R)$ scheme for sharing A_α and B_α and the $(2, 2; M', R')$ scheme for sharing $A_{\bar{\alpha}}$ and $B_{\bar{\alpha}}$. The overall scheme shares A and B with effective cache size $(\alpha M + (1 - \alpha)M')F$ bits and transmission $(\alpha R + (1 - \alpha)R')F$ bits giving a $(2, 2; \alpha M + (1 - \alpha)M', \alpha R + (1 - \alpha)R')$ private scheme. \square

Note that the time sharing scheme in Lemma 2 has a subpacketization that is equal to the sum of the two schemes used for time sharing. Using Lemma 2, and Lemma 1 we can plot the upper bounds for the achievable (M, R) pair for $(2, 2)$ private schemes. The plot is symmetric about the line $M = R$ as can be seen in Fig. 3

C. Towards lower bounds and optimal subpacketization

In the non-private case, the M vs R region is fully characterized for two users and two files. For the case with demand privacy, it is not clear whether any of the points in the achievable M vs R curve shown in Fig. 3 are optimal, or if the subpacketizations are optimal.

While we do not have lower bounds and optimality results yet, we present a few basic impossibility results involving subpacketization and coding of cache contents.

In the non-private case for two users/files, subpacketization of 2 suffices to result in optimal rate of $R = 1/2$ for $M = 1$. For the private case, we have the following result.

Lemma 3. Consider $N = 2$, $K = 2$ with subpacketization of 2 and $M = 1$. A rate $R = 1/2$ cannot be achieved with demand privacy when using a linear scheme.

Proof. A proof is given in Appendix A. \square

For subpacketization of 3, the scheme in Section III-A uses coded cache contents, which is not typical in the non-private setting. In the setting considered here for demand privacy, we have the following result on coding in cache contents.

Lemma 4. Consider $N = 2$, $K = 2$ with subpacketization of 3 and $M = 1$. If the cache contents are not allowed to be coded (i.e. linear combinations of two or more file parts are not allowed to be stored in cache), a rate $R = 2/3$ cannot be achieved with demand privacy when using a linear scheme.

Proof. A proof is given in Appendix B. \square

IV. GENERAL SCHEME AND PARTIAL PRIVACY

In this section, we describe the general scheme from [9] that provides the design of a demand-private coded caching scheme from non-private schemes.

Theorem 5 (Existence of private schemes [9]). *If there exists a $(KN, N; M, R)$ coded caching scheme, then there exists a private $(K, N; M, R)$ scheme.*

Proof. Assume that we have a $(KN, N; M, R)$ non-private scheme. Let the cache contents of each of the users be given as Z'_i , where $0 \leq i < NK$.

Partition the users into sets of size N . Without loss of generality we partition the NK users as

$$\mathcal{U}_k = \{(k-1)N \leq j < kN\}. \quad (3)$$

Denote the cache of the k th user of the private scheme as Z_k . This is chosen as follows:

$$Z_k = Z'_{(k-1)N+r_k} \quad (4)$$

where r_k is uniformly distributed on $\{0, 1, \dots, N-1\}$.

During delivery the server receives the demand vector (d_0, \dots, d_{K-1}) . The server then generates the transmission corresponding to the demand vector of the non-private scheme. This demand vector is of length NK and denoted $D' = (d'_j)$. We can assign any random permutation of the demands $[N]$ to the users in \mathcal{U}_k subject to the condition that $d'_{r_k+(k-1)N} = d_k$. Formally,

$$\pi_k : \mathcal{U}_k \rightarrow [N] \quad (5a)$$

$$d'_j = \pi_k(j) \quad (5b)$$

$$d'_{r_k+(k-1)N} = \pi_k(r_k + (k-1)N) = d_k \quad (5c)$$

Denote the demand vector of the non-private (KN, N) scheme as $D' = (d'_j)_{j \in [NK]}$. Since the non-private scheme can accommodate all demands, it can also serve this demand. Transmit $X^{D'}$ as per the non-private scheme. Then each user of the private scheme is able to receive the file requested.

Demand privacy can be shown as follows. The i -th user of the private scheme is able to recover the file he or she requested. The same transmission can be used to recover all the files by the caches $Z'_{(j-1)K}, \dots, Z'_{jK-1}$. However, the i -th user does not know which of these caches has been assigned to the j -th user. Since all of them are equally likely to be assigned to j -th user by construction, the uncertainty about the demand D_j given D_i, X, Z_i is $H(D_j)$. Thus, the privacy of demands is preserved.

Observe that the cache size of users in the private scheme is same as the size of the cache in the non-private scheme. Similarly, the rate of transmission for the private scheme is exactly the same as that of the non-private scheme. From this it follows the demand private scheme has the parameters $(K, N; M, R)$ as claimed. \square

Remark (Extended demand vector). While creating the extended demand vector D' we can make a simple choice for π_k . The demand of the j th user of the non-private scheme is given as

$$d'_j = d_k - r_k + j \bmod N \text{ for } (k-1)N \leq j < kN, \quad (6)$$

where $0 \leq k < K$.

A. Constructions using Maddah-Ali-Niesen schemes and PDAs

Using the Maddah-Ali-Niesen scheme [12] as the non-private scheme in Theorem 5, we obtain the following:

Corollary 6. *There exists a demand private $(K, N; M, R)$ scheme for integer values of KM , where the rate*

$$R = \begin{cases} \frac{K(N-M)}{(1+KM)} & \text{if } M \geq \frac{K-1}{K} \\ N - M & \text{if } M < \frac{K-1}{K} \end{cases}. \quad (7)$$

Proof. This follows from Theorem 5 using the scheme proposed by Maddah Ali and Niesen [12, Theorem 1]. In this case, for integer values of KM we can construct a $(NK, N; M, R)$ non-private scheme. If $KM \geq K-1$, then $R = \frac{K(N-M)}{(1+KM)}$. If $1 \leq KM < K-1$, then $R = N - M$. We can map each user to a user in the non-private scheme using Eq. (4) and extend the demand vector of the private $(K, N; M, R)$ scheme to the non-private scheme using Eq. (6). Then the scheme from [12] gives the cache contents that should be stored in each

user and a transmission for each demand from which each user can recover their files. The cache memory and rate required in the private scheme will be the same as that in the non-private scheme. \square

Note that there is no coding gain when $KM < K-1$.

A general framework for non-private coded caching schemes was proposed in [27] using placement delivery arrays (PDAs). We can convert many of these schemes to private coded caching schemes. Some of them improve upon those derived from schemes [12] in subpacketization or other parameters. For positive integers K, f, Z and S , a (K, f, Z, S) placement delivery array is a $f \times K$ matrix $(P = [p_{j,k}])$ with $j \in [f], k \in [K]$ containing either a “*” or integers from $\{0, 1, \dots, S-1\}$ in each cell such that they satisfy a few conditions [27]. Here, f is the subpacketization, and S is the total number of transmissions each of size $1/f$ of the file. For any N , we can obtain a $(K, N; \frac{NZ}{f}, \frac{S}{f})$ coded caching scheme from a (K, f, Z, S) placement delivery array.

Corollary 7 (Private schemes from PDAs). *If there exists a (NK, f, Z, S) placement delivery array, we can obtain a private $(K, N; \frac{NZ}{f}, \frac{S}{f})$ coded caching scheme, for any N .*

Proof. Given a (NK, f, Z, S) placement delivery array, there exists a non-private $(NK, N; \frac{NZ}{f}, \frac{S}{f})$ (see [27] for details). From this we can obtain the private $(K, N; \frac{NZ}{f}, \frac{S}{f})$ scheme using Theorem 5. \square

We now present an example of a private scheme with $N = 2, K = 3$, derived from a PDA. Consider the PDA from [27, Eq. (7)] corresponding to 6 users and 4 subfiles.

$$P = \begin{bmatrix} * & 1 & * & 2 & * & 0 \\ 0 & * & * & 3 & 1 & * \\ * & 3 & 0 & * & 2 & * \\ 2 & * & 1 & * & * & 3 \end{bmatrix} \quad (8)$$

We assume that each file W_i is split into f subfiles which are denoted as $W_{i,j}$, where $0 \leq j < f$. In the non-private scheme, the cache contents of the i -th user are given below.

$$\begin{aligned} Z'_0 &= \{W_{i,0}, W_{i,2} : i \in [0, 6)\} \\ Z'_1 &= \{W_{i,1}, W_{i,3} : i \in [0, 6)\} \\ Z'_2 &= \{W_{i,0}, W_{i,1} : i \in [0, 6)\} \\ Z'_3 &= \{W_{i,2}, W_{i,3} : i \in [0, 6)\} \\ Z'_4 &= \{W_{i,0}, W_{i,3} : i \in [0, 6)\} \\ Z'_5 &= \{W_{i,1}, W_{i,2} : i \in [0, 6)\} \end{aligned}$$

The transmission for demand vector $\mathbf{d}' = (d'_0, \dots, d'_5)$ is

$$X^{\mathbf{d}'} = \left\{ \begin{array}{l} W_{d'_0,1} \oplus W_{d'_2,2} \oplus W_{d'_5,0} \\ W_{d'_1,0} \oplus W_{d'_2,3} \oplus W_{d'_4,1} \\ W_{d'_0,3} \oplus W_{d'_3,0} \oplus W_{d'_4,2} \\ W_{d'_1,2} \oplus W_{d'_3,1} \oplus W_{d'_5,3} \end{array} \right\}. \quad (9)$$

For $N = 2$ files, A and B , we can create a private $(3, 2; 1, 1)$ scheme as shown in Fig. 4.

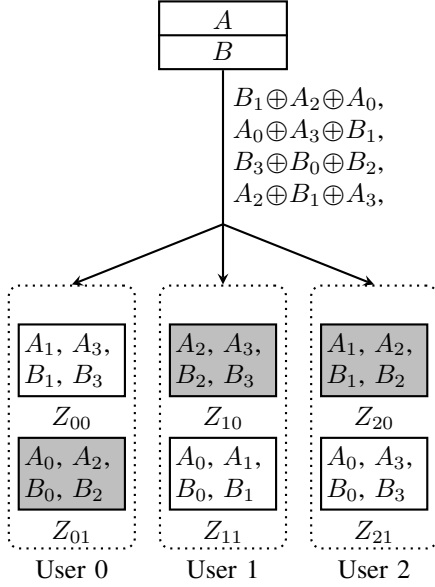


Fig. 4. A $(3, 2; 1, 1)$ private scheme for $D = (A, A, B)$ from a $(6, 2; 1, 1)$ non-private scheme from the PDA given in (8).

B. Case of two files, two users

For the $N = 2$, $K = 2$ case considered earlier, the $M = 1$, $R = 2/3$ construction presented in Section III-A is not derived from a non-private scheme but constructed directly. In fact, a construction from the Maddah-Ali-Niesen scheme using Theorem 5 results in a subpacketization of 6, when compared to the subpacketization of 3 needed for the scheme in Section III-A. This shows that direct construction has the benefits of improved subpacketization.

C. Partial privacy and reduction in subpacketization

The scheme modified from the non-private scheme can have less subpacketization if full privacy is not needed. For instance, suppose that 2-file privacy suffices. That is, at the end of the multicast transmission, every user has an ambiguity of one of two files about any other user's demand.

For 2-file privacy, we need to provide only two options to populate the cache content of a user. Hence, we can use a $(2N, K)$ non-private scheme to arrive at an (N, K) partially private scheme where any user's demand is possibly one of two files to another user. These schemes are important particularly when we have large number of files compared to users. For example, if $N = 10$ and $K = 2$, then a fully private scheme modified from the non-private scheme would require the non-private scheme to have $K' = NK = 20$. With $M = 5$, such a scheme would require a subpacketization $f = \binom{K'}{K} = \binom{20}{2} = 184756$. But under 2-file privacy for this setup, $K' = 2K = 4$, and we can use a subpacketization as low as $f = \binom{4}{2} = 6$. In Fig. 5, we show a partially private $(2, 4; 2, 2/3)$ scheme from a $(4, 4; 2, 2/3)$ non-private scheme providing an ambiguity of two files.

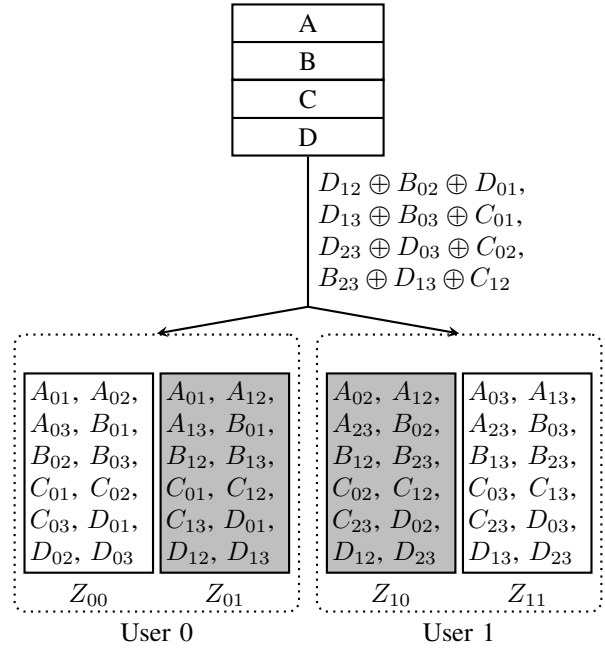


Fig. 5. A $(2, 4; 2, 2/3)$ partially private scheme from a $(4, 4; 2, 2/3)$ non-private scheme. The scheme has a privacy of two files. The gray boxes show the cache assigned by the server. The demands corresponding to unassigned caches for User k are selected at random from the set $\{W_{j,j \in [N]} \setminus D_k\}$. Transmission shown is for the demand vector $D = \{B, D\}$ and the extended demand vector $D = \{D, B, D, C\}$. Observe that cache contents $Z_{00}, Z_{01}, Z_{10}, Z_{11}$ recover the files D, B, D, C , respectively. From the point of view of the User 1, the User 0 could have requested either D or B giving the necessary privacy.

V. CONCLUSION

We have investigated here the problem of demand privacy in systems employing coded caching techniques

with a focus on minimizing subpacketization. For the 2-user, 2-file case, we provided a new construction with a subpacketization of 3. Additionally, we proved that the subpacketization of 3 is indeed minimal for a linear code for the 2-user, 2-file case. Also, we proposed partially private caching schemes and showed how to construct such private schemes with less subpacketization in the general K -user, N -file case.

REFERENCES

- [1] Charu Aggarwal, Joel L Wolf, and Philip S. Yu. Caching on the world wide web. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):94–107, 1999.
- [2] Lee Breslau, Pei Cao, Li Fan, Graham Phillips, Scott Shenker, et al. Web caching and zipf-like distributions: Evidence and implications. In *IEEE INFOCOM '99*, volume 1, pages 126–134. IEEE, 1999.
- [3] Anawat Chankhunthod, Peter B Danzig, Chuck Neerdaels, Michael F Schwartz, and Kurt J Worrell. A hierarchical internet object cache. In *USENIX Annual Technical Conference*, pages 153–164, 1996.
- [4] Zhi Chen, Pingyi Fan, and Khaled Ben Letaief. Fundamental limits of caching: Improved bounds for users with small buffers. *IET Communications*, 10(17):2315–2318, 2016.
- [5] Minquan Cheng, Qifa Yan, Xiaohu Tang, and Jing Jiang. Coded caching schemes with low rate and subpacketizations. *arXiv preprint arXiv:1703.01548*, 2017.
- [6] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 41–50. IEEE, 1995.
- [7] Hooshang Ghasemi and Aditya Ramamoorthy. Improved lower bounds for coded caching. *IEEE Transactions on Information Theory*, 63(7):4388–4413, 2017.
- [8] Jesús Gómez-Vilardebó. Fundamental limits of caching: Improved rate-memory tradeoff with coded prefetching. *IEEE Transactions on Communications*, 66(10):4488–4497, 2018.
- [9] Sneha Kamath. Demand private coded caching. *arXiv preprint arXiv:1909.03324*, 2019.
- [10] Nikhil Karamchandani, Urs Niesen, Mohammad Ali Maddah-Ali, and Suhas N Diggavi. Hierarchical coded caching. *IEEE Transactions on Information Theory*, 62(6):3212–3229, 2016.
- [11] Mohammed Karmoose, Linqi Song, Martina Cardone, and Christina Fragouli. Private broadcasting: an index coding approach. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 2543–2547. IEEE, 2017.
- [12] Mohammad Ali Maddah-Ali and Urs Niesen. Fundamental limits of caching. *IEEE Transactions on Information Theory*, 60(5):2856–2867, 2014.
- [13] Mohammad Ali Maddah-Ali and Urs Niesen. Decentralized coded caching attains order-optimal memory-rate tradeoff. *IEEE/ACM Transactions on Networking (TON)*, 23(4):1029–1040, 2015.
- [14] Scott Michel, Khoi Nguyen, Adam Rosenstein, Lixia Zhang, Sally Floyd, and Van Jacobson. Adaptive web caching: towards a new global caching architecture. *Computer Networks and ISDN systems*, 30(22-23):2169–2177, 1998.
- [15] Urs Niesen and Mohammad Ali Maddah-Ali. Coded caching with nonuniform demands. *IEEE Transactions on Information Theory*, 63(2):1146–1158, 2016.
- [16] Dean Povey and John Harrison. A distributed internet cache. *Australian Computer Science Communications*, 19:175–184, 1997.
- [17] Vaishakh Ravindrakumar, Parthasarathi Panda, Nikhil Karamchandani, and Vinod Prabhakaran. Fundamental limits of secretive coded caching. In *2016 IEEE International Symposium on Information Theory (ISIT)*, pages 425–429. IEEE, 2016.
- [18] Avik Sengupta, Ravi Tandon, and T Charles Clancy. Fundamental limits of caching with secure delivery. *IEEE Transactions on Information Forensics and Security*, 10(2):355–370, 2014.
- [19] Avik Sengupta, Ravi Tandon, and T Charles Clancy. Improved approximation of storage-rate tradeoff for caching via new outer bounds. In *2015 IEEE International Symposium on Information Theory (ISIT)*, pages 1691–1695. IEEE, 2015.
- [20] Chong Shangguan, Yiwei Zhang, and Gennian Ge. Centralized coded caching schemes: A hypergraph theoretical approach. *IEEE Transactions on Information Theory*, 64(8):5755–5766, 2018.
- [21] Karthikeyan Shanmugam, Mingyue Ji, Antonia M Tulino, Jaime Llorca, and Alexandros G Dimakis. Finite-length analysis of caching-aided coded multicasting. *IEEE Transactions on Information Theory*, 62(10):5524–5537, 2016.
- [22] Karthikeyan Shanmugam, Antonia M Tulino, and Alexandros G Dimakis. Coded caching with linear subpacketization is possible using ruzsa-szemerédi graphs. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 1237–1241. IEEE, 2017.
- [23] Li Tang and Aditya Ramamoorthy. Coded caching schemes with reduced subpacketization from linear block codes. *IEEE Transactions on Information Theory*, 64(4):3099–3120, 2018.
- [24] Chao Tian and Jun Chen. Caching and delivery via interference elimination. *IEEE Transactions on Information Theory*, 64(3):1548–1560, 2018.
- [25] Kai Wan and Giuseppe Caire. On coded caching with private demands. *arXiv preprint arXiv:1908.10821*, 2019.
- [26] Kai Wan, Daniela Tuninetti, and Pablo Piantanida. On the optimality of uncoded cache placement. In *2016 IEEE Information Theory Workshop (ITW)*, pages 161–165. IEEE, 2016.
- [27] Qifa Yan, Minquan Cheng, Xiaohu Tang, and Qingchun Chen. On the placement delivery array design for centralized coded caching scheme. *IEEE Transactions on Information Theory*, 63(9):5821–5833, 2017.
- [28] Qifa Yan, Xiaohu Tang, Qingchun Chen, and Minquan Cheng. Placement delivery array design through strong edge coloring of bipartite graphs. *IEEE Communications Letters*, 22(2):236–239, 2017.
- [29] Qian Yu, Mohammad Ali Maddah-Ali, and A Salman Avestimehr. The exact rate-memory tradeoff for caching with uncoded prefetching. *IEEE Transactions on Information Theory*, 64(2):1281–1296, 2017.

APPENDIX A

IMPOSSIBILITY RESULTS FOR $(2, 2)$ PRIVATE LINEAR SCHEMES WITH TWO SUBFILES

A coded caching scheme is said to be linear if all the cache contents, transmissions and the decoding involves only linear operations. Here we provide a proof for Lemma 3 in Section III and show that there does not exist a private $(2, 2; 1, 0.5)$ linear coded caching scheme with subpacketization of two. The proof method is by contradiction. So, we begin by assuming the existence of a $(2, 2; 1, 0.5)$ linear coded caching scheme with subpacketization of two.

A. Notation and setup

Suppose A, B are split into two subfiles each as A_0, A_1 and B_0, B_1 , respectively. Let S be defined as

$$S = \begin{bmatrix} A_0 \\ A_1 \\ B_0 \\ B_1 \end{bmatrix}. \quad (10)$$

Let the i -th user's cache Z_i and the transmission $X^{D_1 D_2}$ be written as

$$Z_i = C_i S, \quad (11)$$

$$X^{D_1 D_2} = T^{D_1 D_2} S, \quad (12)$$

where C_i and $T^{D_1 D_2}$ are 2×4 and 1×4 coefficient matrices, respectively, with entries from a suitable field. User i can decode D_i given $T^{D_1 D_2}$ using the cache Z_i .

The matrix C_i is split into 2×2 matrices C_{iA} and C_{iB} as follows:

$$C_i = \begin{bmatrix} C_{iA} & C_{iB} \end{bmatrix}, \quad (13)$$

Similarly, $T^{D_1 D_2}$ is split into two 1×2 submatrices as shown below.

$$T^{D_1 D_2} = \begin{bmatrix} T_A^{D_1 D_2} & T_B^{D_1 D_2} \end{bmatrix}. \quad (14)$$

We denote the rank of a matrix Q by $\text{rk}(Q)$.

We assume that

$$\text{rk}(C_i) = 2 \quad (15)$$

implying that each cache contains independent subfile combinations.

B. Lemmas on structure of coefficient matrices

Lemma 8 (Rank constraints on coefficient matrices). $\text{rk}(C_{iA}) = \text{rk}(C_{iB}) = 1$.

Proof. We will show $\text{rk}(C_{0B}) = 1$. Consider the 3×4 matrix

$$M_{0AB} = \begin{bmatrix} C_{0A} & C_{0B} \\ T_A^{AB} & T_B^{AB} \end{bmatrix}. \quad (16)$$

The cache of User 0 and transmission for the demand AB , when combined, result in the vector $M_{0AB}S$. Since User 0 can recover $A = [A_0 \ A_1]$ by linearly combining the elements of $M_{0AB}S$, there exists a 2×3 matrix U such that

$$U \begin{bmatrix} C_{0A} \\ T_A^{AB} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (17)$$

and

$$U \begin{bmatrix} C_{0B} \\ T_B^{AB} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad (18)$$

resulting in the decoding of A and elimination of B_0 and B_1 . From Eq. (17), the rank of U is 2. Using this in Eq. (18), $\text{rk}(C_{0B}) \neq 2$.

If C_{0B} is the all-zero matrix, then User 0 cannot recover B using only the transmission X^{BA} . So, $\text{rk}(C_{0B}) \neq 0$. This only leaves the possibility $\text{rk}(C_{0B}) = 1$. The proof above can be readily adapted to show $\text{rk}(C_{iA}) = 1$ for $i = 0, 1$ and $\text{rk}(C_{1B}) = 1$. \square

A consequence of Lemma 8, none of the files are stored entirely on any cache.

For an invertible 2×2 matrix U , the scheme obtained by replacing C_i by UC_i is also a demand-private coded caching scheme because one cache can be obtained from the other. This is captured in the following lemma for future use.

Lemma 9 (Equivalent coefficient matrices). *Cache $Z_i = C_i S$ can recover file W from a transmission X iff $Z'_i = UC_i S$ can recover the same file from X for any invertible 2×2 matrix U .*

Corollary 10 (Reduced coefficient matrices). *Given the coefficient matrix C_i , there exist invertible matrices U_i and V_i such that*

$$V_i U_i C_i = \begin{bmatrix} a & b & 0 & 0 \\ 0 & 0 & c & d \end{bmatrix}, \quad (19)$$

where both (a, b) and (c, d) are nonzero.

Proof. Suppose that C_i is written as follows.

$$C_i = \begin{bmatrix} a & b & * & * \\ a' & b' & * & * \end{bmatrix}. \quad (20)$$

By Lemma 8, $\text{rk}(C_{iA}) = 1$. Hence, without loss of generality we can assume that $(a, b) \neq (0, 0)$, and (a', b') is a scalar multiple of (a, b) . There exists some invertible matrix $U_i = \begin{bmatrix} 1 & 0 \\ \alpha & 1 \end{bmatrix}$ for some scalar α such that

$$U_i C_i = \begin{bmatrix} a & b & c' & d' \\ 0 & 0 & c & d \end{bmatrix} \quad (21)$$

for some c, d, c' and d' . Since $\text{rk}(C_i) = 2$, it follows that $(c, d) \neq (0, 0)$. Then for some β and $V_i = \begin{bmatrix} 1 & \beta \\ 0 & 1 \end{bmatrix}$ we obtain Eq. (19). \square

An immediate consequence of Lemma 9 and Corollary 10, we can assume that the coefficient matrices are of the form given below.

$$C_i = \begin{bmatrix} a_i & b_i & 0 & 0 \\ 0 & 0 & c_i & d_i \end{bmatrix}. \quad (22)$$

Lemma 11 (Constraints due to recovery). *Given $T^{D_0 D_1}$ and C_i , we have the following constraints.*

$$\text{rk} \left(\begin{bmatrix} C_{i D_i} \\ T_{D_i}^{D_0 D_1} \end{bmatrix} \right) = 2 \quad (23a)$$

$$\text{rk} \left(\begin{bmatrix} C_{i \overline{D_i}} \\ T_{\overline{D_i}}^{D_0 D_1} \end{bmatrix} \right) = 1, \quad (23b)$$

where $D_i \in \{A, B\}$ and $\overline{D_i} = \{A, B\} \setminus D_i$ is the file that is not demanded by User i .

Proof. Consider the coefficient matrix C_i of User i , has an equivalent form given in Eq. (22). Combining with $T^{D_0 D_1}$ we have

$$\begin{bmatrix} C_i \\ T^{D_0 D_1} \end{bmatrix} S = \begin{bmatrix} a_i & b_i & 0 & 0 \\ 0 & 0 & c_i & d_i \\ T_A^{D_0 D_1} & T_B^{D_0 D_1} \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ B_0 \\ B_1 \end{bmatrix}. \quad (24)$$

From this system of equations, one can observe that A_0 and A_1 appear in two equations. If $D_i = A$, then User i must recover the subfiles A_0 and A_1 , and the following condition must hold.

$$\text{rk} \left(\begin{bmatrix} a_i & b_i \\ T_A^{D_0 D_1} \end{bmatrix} \right) = 2 \quad (25)$$

Similarly if $D_i = B$, the following condition must hold for User i to recover file B from $T^{D_0 D_1}$.

$$\text{rk} \left(\begin{bmatrix} c_i & d_i \\ T_B^{D_0 D_1} \end{bmatrix} \right) = 2. \quad (26)$$

Eq. (23a) follows from Eq. (25) and Eq. (26).

One can see that the condition in Eq. (25) is not enough for recovering $D_i = A$ at User i using Eq. (24). We should be able to remove the part corresponding to $T_{\overline{D_i}}^{D_0 D_1} = T_B^{D_0 D_1}$ from the third row in Eq. (24) to arrive at two equations in two variables A_0, A_1 and solve for them. So if $T_B^{D_0 D_1}$ is nonzero, then it should be a scalar multiple of (c_i, d_i) . Since (c_i, d_i) is nonzero from Lemma 8, we have

$$\text{rk} \left(\begin{bmatrix} c_i & d_i \\ T_B^{D_0 D_1} \end{bmatrix} \right) = 1 \quad (27)$$

Hence

$$\text{rk} \left(\begin{bmatrix} C_{0B} \\ T_B^{AD_1} \end{bmatrix} \right) = 1 \text{ and } \text{rk} \left(\begin{bmatrix} C_{1B} \\ T_B^{D_0 A} \end{bmatrix} \right) = 1. \quad (28)$$

Similarly solving for B_0 and B_1 (i.e. $D_i = B$) at User i requires

$$\text{rk} \left(\begin{bmatrix} a_i & b_i \\ T_A^{D_0 D_1} \end{bmatrix} \right) = 1 \quad (29)$$

Hence

$$\text{rk} \left(\begin{bmatrix} C_{0A} \\ T_A^{BD_1} \end{bmatrix} \right) = 1 \text{ and } \text{rk} \left(\begin{bmatrix} C_{1A} \\ T_A^{D_0 B} \end{bmatrix} \right) = 1. \quad (30)$$

Eq. (28) and Eq. (30) immediately imply Eq. (23b). \square

So far, we have not used the requirement of demand privacy. The following lemma uses the demand privacy condition to derive an important constraint on the transmission.

Lemma 12 (Constraints on transmission). *If $X^{D_0 D_1} = T^{D_0 D_1} S$ where S is defined as in Eq. (10), then T_A^{AA} and T_B^{AA} are both nonzero.*

Proof. If T_A^{AA} is zero, then User 0 cannot recover file A . So, T_A^{AA} is nonzero. We know that the entire file B is not stored on any cache. If T_B^{AA} is zero, then every user must be demanding only A . This reveals the demands of all the users, so T_B^{AA} must be nonzero. \square

Note that Lemma 12 is only a necessary condition for demand privacy.

C. Proof of Lemma 3

Let the coefficient matrix $T^{AA} = (u, v, w, x)$. From Lemma 12, $(u, v) \neq 0$ and $(w, x) \neq 0$.

$$\begin{bmatrix} C_i \\ T^{AA} \end{bmatrix} S = \begin{bmatrix} a_i & b_i & 0 & 0 \\ 0 & 0 & c_i & d_i \\ u & v & w & x \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ B_0 \\ B_1 \end{bmatrix} \quad (31)$$

By Eq. (23b), we have

$$\text{rk} \left(\begin{bmatrix} c_i & d_i \\ w & x \end{bmatrix} \right) = 1$$

Both (c_i, d_i) and (w, x) are nonzero due to Lemma 8 and Lemma 12 respectively. Thus, both (w, x) and (c_i, d_i) are scalar multiples of each other for $i = 0, 1$. This implies that (c_0, d_0) is a scalar multiple of (c_1, d_1) . Then $\text{rk} \left(\begin{bmatrix} c_i & d_i \\ T_B^{AB} \end{bmatrix} \right)$ is same for $i = 0, 1$. However, this contradicts Lemma 11, by which

$$\text{rk} \left(\begin{bmatrix} c_0 & d_0 \\ T_B^{AB} \end{bmatrix} \right) = 1 \text{ and } \text{rk} \left(\begin{bmatrix} c_1 & d_1 \\ T_B^{AB} \end{bmatrix} \right) = 2.$$

This shows the in-feasibility of coefficient matrices satisfying the rank constraints due to recovery and demand privacy.

Therefore, we conclude that a linear private $(2, 2; 1, 1/2)$ coded caching scheme with subpacketization of two does not exist.

APPENDIX B
IMPOSSIBILITY RESULTS FOR UNCODED
PREFETCHING WITH THREE SUBFILES

In Appendix A, we have seen that with two subfiles we cannot obtain a private $(2, 2; 1, 1/2)$ scheme. Here we show that without coded prefetching we cannot obtain a private $(2, 2; 1, 2/3)$ scheme with three subfiles and thereby prove Lemma 4.

Informally, the proof is organized as follows. First, we show that without coded prefetching the subfiles must be cached in an uncoded form i.e. without linear combinations. This restricts the possibilities for the caches. Furthermore any given cache restricts the possibilities for the other user's cache. Demand privacy is possible only if the set of caches consistent with a user allow the reconstruction of both the files for any demand. We show that is not possible and hence a linear private $(2, 2; 1, 2/3)$ scheme with subpacketization of three subfiles does not exist.

A. Permissible caches without coded prefetching

Without coded prefetching, the subfiles can only be replicated in the cache. With three subfiles, $M = 1$ implies that each user can store 3 subfiles. $R = 2/3$ implies that there are two independent subfile combinations in the transmission. If all the subfiles in a cache belongs to a file, that user cannot recover the other file from a transmission of rate $R = 2/3$. So a cache should contain two subfiles of one file and one subfile of the other file. Let the two files be A and B. Without loss of generality, let us assume the cache of first user, Z_0 contains two subfiles of file A and one subfile of B.

$$Z_0 = \{A_0, A_1, B_2\}. \quad (32)$$

Let the cache of User 1 be

$$Z_1 = \{G_0, G_1, G_2\}, \quad (33)$$

where $G_i \in \{A_0, A_1, A_2, B_0, B_1, B_2\}$.

Lemma 13. *If $Z_0 = \{A_0, A_1, B_2\}$, then the permissible cache for Z_1 must be one of the following.*

$$Z_1 = \{G_0, G_1, A_2 \mid G_0, G_1 \in \{B_0, B_1, B_2\}\} \quad (34a)$$

$$\text{or } Z_1 = \left\{ G_0, G_1, A_2 \mid \begin{array}{l} G_0 \in \{A_0, A_1\} \\ G_1 \in \{B_0, B_1, B_2\} \end{array} \right\}. \quad (34b)$$

Proof. Consider the transmission

$$X^{BA} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \alpha_0 A_0 + \alpha_1 A_1 + \alpha_2 A_2 + \beta_0 B_0 + \beta_1 B_1 + \beta_2 B_2 \\ \gamma_0 A_0 + \gamma_1 A_1 + \gamma_2 A_2 + \delta_0 B_0 + \delta_1 B_1 + \delta_2 B_2 \end{bmatrix} \quad (35)$$

User 0 can use its cache contents to eliminate three variables from the system of linear equations in Eq. (35). The reduced/equivalent equations for User 0 is

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} \alpha_2 A_2 + \beta_0 B_0 + \beta_1 B_1 \\ \gamma_2 A_2 + \delta_0 B_0 + \delta_1 B_1 \end{bmatrix}$$

Since User 0 does not have access to A_2 , for recovering B_0 and B_1 we need

$$\text{rk} \left(\begin{bmatrix} \beta_0 & \beta_1 \\ \delta_0 & \delta_1 \end{bmatrix} \right) = 2 \text{ and } \begin{bmatrix} \alpha_2 \\ \gamma_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (36)$$

The transmission X^{BA} cannot involve A_2 . So A_2 must be in Z_1 for it to recover file A from X^{BA} .

$$G_2 = A_2$$

All the subfiles in Z_1 cannot be that of file A. So, we have two cases for the possible values of $\{G_0, G_1\}$ based on the associated files. Either both of them are subfiles of B as in Eq. (34a) or one of them is subfile of A and the other is of B as in Eq.(34b). \square

B. Two subfiles of file B in Z_1

In this section, we will show that if the cache of User 1 is of the form given in Eq. (34a), then the scheme is not private.

Lemma 14. *If $Z_0 = \{A_0, A_1, B_2\}$, and $Z_1 = \{G_0, G_1, A_2\}$ and $G_i \in \{B_0, B_1, B_2\}$, then demand privacy is not satisfied.*

Proof. Let $G_0, G_1 \in \{B_0, B_1, B_2\}$. The reduced equations corresponding to X^{BA} for User 1 will be

$$\begin{bmatrix} u'' \\ v'' \end{bmatrix} = \begin{bmatrix} \alpha_0 A_0 + \alpha_1 A_1 + \beta_0 B_0 + \beta_1 B_1 + \beta_2 B_2 \\ \gamma_0 A_0 + \gamma_1 A_1 + \delta_0 B_0 + \delta_1 B_1 + \delta_2 B_2 \end{bmatrix} \quad (37)$$

For User 1 being able to obtain subfiles A_0 and A_1 from the transmission, we need

$$\text{rk} \left(\begin{bmatrix} \alpha_0 & \alpha_1 \\ \gamma_0 & \gamma_1 \end{bmatrix} \right) = 2, \text{ and} \quad (38a)$$

$$\begin{bmatrix} \beta_2 \\ \delta_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (38b)$$

Due to Eq. (36), Z_2 must contain the subfiles B_0 and B_1 , for eliminating those variables from X^{BA} and recover the subfiles of A.

$$\{G_0, G_1\} = \{B_0, B_1\} \quad (39)$$

Now consider the transmission for $D = (A, A)$.

$$X^{AA} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \alpha'_0 A_0 + \alpha'_1 A_1 + \alpha'_2 A_2 + \beta'_0 B_0 + \beta'_1 B_1 + \beta'_2 B_2 \\ \gamma'_0 A_0 + \gamma'_1 A_1 + \gamma'_2 A_2 + \delta'_0 B_0 + \delta'_1 B_1 + \delta'_2 B_2 \end{bmatrix} \quad (40)$$

For User 1, reduced equations are

$$\begin{bmatrix} u'' \\ v'' \end{bmatrix} = \begin{bmatrix} \alpha'_0 A_0 + \alpha'_1 A_1 + \beta'_2 B_2 \\ \gamma'_0 A_0 + \gamma'_1 A_1 + \delta'_2 B_2 \end{bmatrix} \quad (41)$$

For User 1 recovering A_0 and A_1 , it requires

$$\text{rk} \left(\begin{bmatrix} \alpha'_0 & \alpha'_1 \\ \gamma'_0 & \gamma'_1 \end{bmatrix} \right) = 2 \text{ and} \quad (42a)$$

$$\begin{bmatrix} \beta'_2 \\ \delta'_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (42b)$$

For demand privacy we require the existence of some cache Z'_1 which can recover file B from X^{AA} . Since X^{AA} doesn't involve B_2 , it must be present in Z' .

$$Z'_1 = \{H_0, B_2, A_2\}$$

For no value of $H_0 \in \{B_0, B_1, A_0, A_1\}$, it can recover both B_0 and B_1 (or file B completely) from X^{AA} due to Eq. (42a). Thus, if Z_0 has two subfiles of A , Z_1 cannot contain two subfiles of B as given in Eq. (34a). \square

C. Two subfiles of file A in Z_1

If the cache of User 1 is of the form given in Eq. (34b), then we can restrict the cache even further as the following lemma shows.

Lemma 15. *If $Z_0 = \{A_0, A_1, B_2\}$, then the permissible cache for Z_1 must be of the form $Z_1 = \{G_0, G_1, A_2\}$, where G_0, G_1 are distinct and $G_0 \in \{A_0, A_1\}$ and $G_1 \in \{B_0, B_1\}$.*

Proof. We need to show $G_1 \neq B_2$. Since Z_1 already contains A_2 , it can have either A_0 or A_1 , both of which are in Z_0 . Without loss of generality, let $G_0 = A_1$. Assume $G_1 = B_2$. Then $Z_1 = \{B_2, A_1, A_2\}$. Consider the transmission

$$X^{AB} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \alpha_0 A_0 + \alpha_1 A_1 + \alpha_2 A_2 + \beta_0 B_0 + \beta_1 B_1 + \beta_2 B_2 \\ \gamma_0 A_0 + \gamma_1 A_1 + \gamma_2 A_2 + \delta_0 B_0 + \delta_1 B_1 + \delta_2 B_2 \end{bmatrix} \quad (43)$$

For User 0, these equations reduces to

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} \alpha_2 A_2 + \beta_0 B_0 + \beta_1 B_1 \\ \gamma_2 A_2 + \delta_0 B_0 + \delta_1 B_1 \end{bmatrix} \quad (44)$$

Since User 0 have no access to B_0 and B_1 , for recovering A_2 , we need

$$\text{rk} \left(\begin{bmatrix} \beta_0 & \beta_1 \\ \delta_0 & \delta_1 \end{bmatrix} \right) \leq 1 \text{ and} \quad (45a)$$

$$\begin{bmatrix} \alpha_2 \\ \gamma_2 \end{bmatrix} \neq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (45b)$$

For User 1 the equations from X^{AB} reduces to

$$\begin{bmatrix} u'' \\ v'' \end{bmatrix} = \begin{bmatrix} \alpha_0 A_0 + \beta_0 B_0 + \beta_1 B_1 \\ \gamma_0 A_0 + \delta_0 B_0 + \delta_1 B_1 \end{bmatrix} \quad (46)$$

For User 1 recovering B_0 and B_1 , it requires

$$\text{rk} \left(\begin{bmatrix} \beta_0 & \beta_1 \\ \delta_0 & \delta_1 \end{bmatrix} \right) = 2 \text{ and} \quad (47a)$$

$$\begin{bmatrix} \alpha_0 \\ \gamma_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (47b)$$

Equations (47a) and (45a) are contradictory, Hence $G_1 \neq B_2$. So $G_1 \in \{B_0, B_1\}$. \square

By Lemma 15, there are four possible choices for Z_1 as given below.

$$Z_a = \{A_1, A_2, B_0\} \quad (48a)$$

$$Z_b = \{A_1, A_2, B_1\} \quad (48b)$$

$$Z_c = \{A_0, A_2, B_0\} \quad (48c)$$

$$Z_d = \{A_0, A_2, B_1\} \quad (48d)$$

Lemma 16. *If $Z_0 = \{A_0, A_1, B_2\}$ and $Z_1 \in \{Z_a, Z_b, Z_c, Z_d\}$, then demand privacy is not possible.*

Proof. It suffices to show demand privacy is not possible for $Z_1 = Z_a$, since we can arrive at the other cache combinations by relabeling.

Suppose the cache $Z_a = \{A_1, A_2, B_0\}$ is assigned to the User 1. Then, by arguments similar to Lemmas 13, 14 and 15, the User 1 is aware that the cache of User 0 must have two subfiles of A , with one being A_0 and the subfile of B in Z_0 is not B_0 . The four possible caches for Z_0 consistent with $Z_1 = Z_a$ are given below.

$$Z_e = \{A_0, A_1, B_1\} \quad (49a)$$

$$Z_f = \{A_0, A_1, B_2\} \quad (49b)$$

$$Z_g = \{A_0, A_2, B_1\} \quad (49c)$$

$$Z_h = \{A_0, A_2, B_2\} \quad (49d)$$

Note that $Z_0 = Z_f$. For demand privacy we need the caches consistent with Z_0 to be able to recover both files and vice versa.

Consider the transmission

$$X^{AB} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \alpha_0 A_0 + \alpha_1 A_1 + \alpha_2 A_2 + \beta_0 B_0 + \beta_1 B_1 + \beta_2 B_2 \\ \gamma_0 A_0 + \gamma_1 A_1 + \gamma_2 A_2 + \delta_0 B_0 + \delta_1 B_1 + \delta_2 B_2 \end{bmatrix} \quad (50)$$

For the User 0 with cache $Z_0 = \{A_0, A_1, B_2\}$ the transmission X^{AB} reduces to the following set of equations after eliminating the subfiles which are already present in Z_0 .

$$\begin{bmatrix} u_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} \alpha_2 A_2 + \beta_0 B_0 + \beta_1 B_1 \\ \gamma_2 A_2 + \delta_0 B_0 + \delta_1 B_1 \end{bmatrix} \quad (51)$$

For User 0 whose demand is A already has A_0 and A_1 . Only A_2 needs to be recovered from X^{AB} . This is possible only if the following conditions are satisfied.

$$\text{rk} \left(\begin{bmatrix} \beta_0 & \beta_1 \\ \delta_0 & \delta_1 \end{bmatrix} \right) \leq 1 \text{ and} \quad (52a)$$

$$\begin{bmatrix} \alpha_2 \\ \gamma_2 \end{bmatrix} \neq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (52b)$$

Similarly, for the User 1, whose cache is $Z_1 = Z_a$, the transmission X^{AB} reduces to

$$\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} = \begin{bmatrix} \alpha_0 A_0 + \beta_1 B_1 + \beta_2 B_2 \\ \gamma_0 A_0 + \delta_1 B_1 + \delta_2 B_2 \end{bmatrix} \quad (53)$$

For User 1 to recover B_0 and B_1 , the following conditions must be satisfied.

$$\text{rk} \left(\begin{bmatrix} \beta_1 & \beta_2 \\ \delta_1 & \delta_2 \end{bmatrix} \right) = 2 \text{ and} \quad (54a)$$

$$\begin{bmatrix} \alpha_0 \\ \gamma_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (54b)$$

The reduced equations for Z_b are

$$\begin{bmatrix} u_b \\ v_b \end{bmatrix} = \begin{bmatrix} \alpha_0 A_0 + \beta_0 B_0 + \beta_2 B_2 \\ \gamma_0 A_0 + \delta_0 B_0 + \delta_2 B_2 \end{bmatrix} \quad (55)$$

The reduced equations for Z_c are

$$\begin{bmatrix} u_c \\ v_c \end{bmatrix} = \begin{bmatrix} \alpha_1 A_1 + \beta_1 B_1 + \beta_2 B_2 \\ \gamma_1 A_1 + \delta_1 B_1 + \delta_2 B_2 \end{bmatrix} \quad (56)$$

The reduced equations for $Z_d = Z_g$ are

$$\begin{bmatrix} u_d \\ v_d \end{bmatrix} = \begin{bmatrix} \alpha_1 A_1 + \beta_0 B_0 + \beta_2 B_2 \\ \gamma_1 A_1 + \delta_0 B_0 + \delta_2 B_2 \end{bmatrix} \quad (57)$$

The reduced equations for Z_e are

$$\begin{bmatrix} u_e \\ v_e \end{bmatrix} = \begin{bmatrix} \alpha_2 A_2 + \beta_0 B_0 + \beta_2 B_2 \\ \gamma_2 A_2 + \delta_0 B_0 + \delta_2 B_2 \end{bmatrix} \quad (58)$$

The reduced equations for Z_h are

$$\begin{bmatrix} u_h \\ v_h \end{bmatrix} = \begin{bmatrix} \alpha_1 A_1 + \beta_0 B_0 + \beta_1 B_1 \\ \gamma_1 A_1 + \delta_0 B_0 + \delta_1 B_1 \end{bmatrix} \quad (59)$$

From the above constraints and the reduced equations for all users we can infer the following.

- 1) Due to Eq. (54b), Z_b cannot recover file A since it has no access to A_0 .
- 2) Due to Eq. (54a), and since Z_c has no access to B_1 and B_2 , it cannot eliminate them from the transmission to recover file A .
- 3) Due to Eq. (52b), Z_e cannot recover file B .
- 4) Due to Eq. (52a), Z_h cannot recover file B .

Hence, from the four possible caches for Z_1 , the only cache that might be able to recover file A and might achieve privacy for User 1 is Z_d . But since there are only five equations from the cache and transmissions, it is impossible for Z_d to recover all the six subfiles $A_0, A_1, A_2, B_0, B_1, B_2$ and thus recover file B also. That means, no possible cache for User 0 consistent with Z_1 is able to recover file B from X^{AB} . This results in no demand privacy for User 0.

On the other hand, if X^{AB} is such that Z_d can recover file B , then it results in no privacy for User 1. \square

Note that any consistent set of caches for User 0 and User 1 can be obtained by permuting subfile labels of file A and file B (permutation π_A to relabel A and π_B to relabel B). Applying the same relabeling, the above proof will hold true for them as well. This concludes the proof of Lemma 4.