Gurjarpadhye, C., Ravi, J., Dey, B. K. & Karamchandani, N. (2021). *Improved Memory-Rate Trade-off for Caching with Demand Privacy*. In: 2020 IEEE Information Theory Workshop (ITW), 11-15 April, 2020.

DOI: 10.1109/itw46852.2021.9457647

Improved Memory-Rate Trade-off for Caching with Demand Privacy

Chinmay Gurjarpadhye*, Jithin Ravi†, Bikash Kumar Dey*, Nikhil Karamchandani*
* Indian Institute of Technology Bombay, Mumbai
† Universidad Carlos III de Madrid, Leganés, Spain
Emails: cgurjarpadhye@gmail.com, rjithin@tsc.uc3m.es, {bikash, nikhilk}@ee.iitb.ac.in

***Abstract*—We consider the demand-private coded caching problem in a noiseless broadcast network. It is known from past works that a demand-private scheme for N files and K users can be obtained from a non-private scheme for N files and NK users. We first propose a scheme that improves on this idea by removing some redundant transmissions. The memory-rate trade-off achieved using this scheme is shown to be within a multiplicative factor of 3 from the optimal for all the memory regimes when $K < N$. We further show that a demand-private scheme for N files and K users can be obtained from a particular known non-private scheme for N files and $NK \quad K+1$ users. Finally, we give the exact memory-rate trade-off for demand-private coded caching problems with $N > K = 2$.**

I. Introduction

In the seminal work [1], Maddah-Ali and Niesen analyzed the fundamental limits of caching in an error-free broadcast network from an information-theoretic perspective. A server has N files. There are K users, each equipped with a cache that can store M files. In the *placement phase*, the cache of each user is populated with some functions of the files. In the *delivery phase*, each user requests one of the N files, and the server broadcasts a message to serve the demands of the users. The goal of the coded caching problem is to reduce the *rate* of transmission from the server for a given cache size M. For the successful decoding of the files, the scheme proposed in [1] requires the demand vector to be known globally.

In this paper, we consider the coded caching problem with an extra constraint that each user should not learn any information about the demands of other users. Coded caching under demand privacy was studied from an information-theoretic framework in [2]–[6]. The works [2], [3] demonstrated that a demand-private scheme for N files and K users can be obtained from a non-private scheme for N files and NK users. A new demand-private scheme was constructed in [4] and combined with previous results, the achievable rate was shown to be within a constant factor of the information-theoretic optimal. In fact [4] showed that the additional cost of privacy is bounded in the sense that the optimal rates with and without demand privacy are within a constant multiplicative factor. Furthermore, the exact *memory-rate trade-off* for $N = K = 2$ was also obtained in [4]. In [5], the authors focused on obtaining demand-private schemes that achieve a weaker privacy condition such that one user should not get any information about the demand of another user, but may gain some information about the demand vector. They mainly addressed the *subpacketization* requirement for $N = K = 2$. Demand privacy against colluding users was studied for device-to-device network in [7] where a trusted server helps to co-ordinate among the users to achieve a demand-private scheme. The case of colluding users for the coded caching problem was considered in [6]. Since the users may collude, the privacy condition in [6] was such that one user should not learn any information about the demands of other users even if she is given all the files.

Ravindrakumar *et al.* [8] investigated the *privacy of files* for the coded caching problem, i.e., each user should not get any information about any file other than the requested one. A private scheme was proposed using the techniques from secret sharing. Sengupta *et al.* [9] investigated the privacy of files against an eavesdropper who has access to the broadcast link.

In this paper, we study demand-private caching, i.e., the demands of the other users remain information-theoretically private from each user. In [4], it was shown that one can obtain a demand-private scheme for N files and K users from a non-private scheme for N files and NK users which serves only a subset of demand vectors. We first propose a scheme (Scheme A) that builds on this fact. Memory-rate pairs achievable using Scheme A are given in Theorem 1. The memory-rate pairs in Theorem 1 are shown to be within a multiplicative gap of 3 from the optimal for $K < N$ (Theorem 2). This improves state of the art on order optimality result from [4] where a gap of 8 was shown.

We construct a demand-private scheme (Scheme B) for N files and K users from the non-private scheme for N files and $NK - K + 1$ users proposed in [10] (which we refer to as the YMA scheme). The achievable rates using Scheme B are better than the ones achievable using Scheme A for $N \leq K$, whereas the opposite is true for $N > K$.

One class of instances for which the exact trade-off is known for non-private schemes [1], [11] is when $K = 2$ and $N \geq 2$. The exact memory-rate trade-off under demand privacy for $N = K = 2$ was characterized in [4]. In Theorem 4, we characterize the exact trade-off under demand privacy for $N >$

J. Ravi has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant No. 714161). The work of B. K. Dey was supported in part by the Bharti Centre for Communication in IIT Bombay. The work of N. Karamchandani is supported in part by a Science and Engineering Research Board (SERB) grant on "Content Caching and Delivery over Wireless Networks".

2 and $K = 2$. To this end, we present two novel achievability schemes. Then, by generalizing the converse bound presented in [4], we demonstrate that the achievable region given by these schemes is in fact the exact memory-rate trade-off.

The rest of the paper is organized as follows. In Section II, we give our problem formulation. We present Scheme A and Scheme B in Section III. Finally, in Section IV, we give the exact trade-off for $N > 2$ and $K = 2$.

Notations: We denote the set $\{0, 1, \ldots, N-1\}$ by[1] $[N]$, the cardinality of a set $\mathcal{A}$ by $|\mathcal{A}|$, and the closed interval between two real numbers a and b by $[a, b]$. For a positive integer ℓ, if π denotes a permutation of $[\ell]$, and $Y = (Y_0, Y_1, \cdots, Y_{\ell-1})$, with abuse of notation, we define $\pi(Y) = \left(Y_{\pi^{-1}(i)}\right)_{i\in[\ell]}$. We denote random variables by upper case letters (e.g. X) and their alphabets by calligraphic letters (e.g. $\mathcal{X}$). For a random variable/vector B, $len(B)$ denotes $\log_2 |\mathcal{B}|$.

II. Problem formulation and definitions

Consider one server connected to K users through a noiseless broadcast link. The server has access to N independent files of F bits each. These files are denoted as $(W_0, W_1, \cdots, W_{N-1})$ and each file is uniformly distributed in $\{0, 1\}^F$. Each user has a cache of size MF bits. The coded caching problem has two phases: prefetching and delivery. In the prefetching phase, the server places at most MF bits in the cache of each user. The cache content of user $k \in [K]$ is denoted by Z_k. In the delivery phase, each user demands one of the N files from the server and this demand is conveyed secretly to the server. Let the demand of user k be denoted by $D_k \in [N]$. We define $\bar{D} = (D_0, D_1, \cdots, D_{K-1})$. $\bar{D}$ is independent of the files $W_i; i \in [N]$ and caches $Z_k; k \in [K]$, and is uniformly distributed in $[N]^K$.

In the delivery phase, the server broadcasts a message X to all the K users such that user $k \in [K]$ can decode file W_{D_k} using X and Z_k. If message X consists of RF bits, then R is said to be the rate of transmission. In addition to the decodability of the demanded file, demand-privacy imposes another constraint that the demands of all other users should remain perfectly secret to each of the K users. To ensure demand-privacy the server can share some randomness denoted by S_k with user $k \in [K]$ in the prefetching phase. This shared randomness is of negligible size and hence, it is not included in the memory size. We define $S = (S_0, S_1, \cdots, S_{K-1})$. The server also has access to some private randomness which we denote by P. The random variables $S, P, \{W_i | i \in [N]\}, \{D_k | k \in [K]\}$ are independent of each other.

Any (N, K, M, R) demand-private coded caching scheme consists of broadcast encoding functions E and J, and a cache encoding function C_k and a decoding function G_k for each user $k \in [K]$. The cache encoding function

$$C_k : [2^F]^N \times \mathcal{S}_k \times \mathcal{P} \to [2^{MF}]$$

gives the cache content $Z_k = (C_k(\bar{W}, S_k, P), S_k)$. The broadcast encoding functions

$$E : [2^F]^N \times \mathcal{D}_0 \times \mathcal{D}_1 \times \cdots \times \mathcal{D}_{N-1} \times \mathcal{S} \times \mathcal{P} \to [2^{RF}],$$

and $J : \mathcal{D}_0 \times \mathcal{D}_1 \times \cdots \times \mathcal{D}_{N-1} \times \mathcal{S} \times \mathcal{P} \to \mathcal{J}$ output the broadcasted message $X = (E(\bar{W}, \bar{D}, S, P),\ J(\bar{D}, S, P))$. The auxiliary transmission $J(\bar{D}, S, P))$ is chosen such that $\log_2 |\mathcal{J}|$ is $o(F)$. Thus, the size of $J(\bar{D}, S, P)$ is negligible and does not contribute to the rate. The decoding function

$$G_k : [2^{MF}] \times \mathcal{S}_k \times \mathcal{D}_k \times \mathcal{J} \times [2^{RF}] \to [2^F]$$

decodes W_{D_k} at user k, i.e.,

$$W_{D_k} = G_k(Z_k, X, D_k) \tag{1}$$

which holds for all realizations of $\bar{D}$, $\bar{W}$, S and P. Along with decodability, the following privacy constraint must be satisfied for any (N, K, M, R) demand-private scheme:

$$I(\bar{D}_{\tilde{k}}; Z_k, D_k, X) = 0 \tag{2}$$

where $\bar{D}_{\tilde{k}} = (D_0, \cdots, D_{k-1}, D_{k+1}, \cdots, D_{K-1})$. The optimal rate for a demand-private coded caching problem of N files, K users and memory M is denoted by $R^{*p}_{N,K}(M)$ and defined as

$$R^{*p}_{N,K}(M) = \inf\{R :\ \text{there exitsts an } (N, K, M, R) \text{ demand-private scheme}\}.$$

For a given M, the pair $(M, R^{*p}_{N,K}(M))$ is called the exact memory-rate trade-off under demand privacy.

An (N, K, M, R) non-private scheme is defined similarly. We do not need shared randomness and private randomness, i.e., $\mathcal{S}_k = \phi, k \in [K]$ and $\mathcal{P} = \phi$. The auxiliary transmission is simply the demand vector, i.e., $J = \bar{D}$. The privacy condition is absent in the non-private scheme and we only have the decodability condition (1). The optimal rate for a non-private scheme is denoted by $R^{*}_{N,K}(M)$ and can be defined similarly as $R^{*p}_{N,K}(M)$. More details on the definition of a non-private coded caching scheme can be found in [4].

III. Achievability Schemes and their performance

In this section, we propose two novel demand-private schemes, Scheme A and Scheme B. Scheme A outperforms Scheme B for $K < N$, and vice-versa for $N \leq K$. Also as shown in Theorem 2, using Scheme A the known order optimality factor for $K < N$ is improved.

A. Achievable Memory-Rate Pairs using Scheme A

As shown in prior works [2]–[4], an (N, K)-private scheme can be obtained from an (N, NK)-non-private scheme. We build on the ideas in these schemes and propose Scheme A. The memory-rate pairs achievable using Scheme A is given in (3) and (4) in Theorem 1. Furthermore, the achievability of memory-rate pairs $(0, K)$ and $(N, 0)$ with demand privacy was shown in [4]. Thus, we have the following theorem.

[1]This notation is different from the more commonly used notation, where $[N]$ denotes the set $\{1, 2, 3, \ldots, N\}$.

Theorem 1 *There exist* (N, K, M, R)*-private schemes achieving the memory-rate pairs* $(0, K)$*,* $(N, 0)$ *and also* (M, R) *pairs with*

$$M = \frac{N\sum_{s=t}^{NK-1}\binom{NK-1}{s-1}r^{NK-s-1}}{\sum_{s=t}^{NK-1}\binom{NK}{s}r^{NK-s-1}}, \tag{3}$$

$$R = \frac{\sum_{s=t+1}^{NK}[\binom{NK}{s}-\binom{NK-K}{s}]r^{NK-s}}{\sum_{s=t}^{NK-1}\binom{NK}{s}r^{NK-s-1}}, \tag{4}$$

for $t = \{1, \ldots, NK-1\}$*,* $r \in [1, N-1]$*.*

Note that for the non-zero memory-rate pairs in Theorem 1, we have 2 free parameters t and r. By fixing the value of r, one can obtain a memory-rate curve by varying the value of t. We have observed through simulations that the memory-rate curve achieved for $r = r_1$ is better than that for $r = r_2$ if $r_1 > r_2$. The memory-rate curve for $r < N-1$, although empirically suboptimal compared to $r = N-1$, is useful in obtaining the result on order optimality in Theorem 2. In the next subsection, we illustrate the scheme which achieves the non-zero memory-rate pairs in Theorem 1 using an example. The general scheme can be found in the extended version [12].

B. An Example of Scheme A for $(N, K, M) = (3, 2, \frac{195}{116})$

Now we illustrate the scheme that achieves the non-zero memory-rate pairs in Theorem 1 for $N = 3, K = 2$. Let us consider the demand-private coded caching problem for 3 files and 2 users. By choosing $r = 2$ and $t = 3$ in the expression for (M, R) in Theorem 1, we get $M = \frac{195}{116}$ and $R = \frac{69}{116}$. Next we describe the scheme which achieves this memory-rate pair with $F = 116l$ for some positive integer l. We partition each file $W_i, i \in [3]$ into $\sum_{j=t}^{NK-1}\binom{NK}{j} = \sum_{j=3}^{5}\binom{6}{j} = 41$ segments of three different sizes. These segments are grouped into three groups such that all segments in one group have the same size. The segments are labelled by some subsets of $[NK] = [6]$. The segments of W_i are $W_{i,\mathcal{R}}$; $\mathcal{R} \subset [6], |\mathcal{R}| = 3, 4, 5$. These segments are of different sizes, and these are grouped into 3 groups as

$$\mathcal{T}_j^i = (W_{i,\mathcal{R}})_{\mathcal{R}\subset[6],|\mathcal{R}|=j}, \quad \text{for } j = 3, 4, 5.$$

The size of segment $W_{i,\mathcal{R}}, i \in [3]$ is $len(W_{i,\mathcal{R}}) = r^{|\mathcal{R}|-NK+1}l = r^{|\mathcal{R}|-5}l$. Thus, each segment in $\mathcal{T}_5^i, \mathcal{T}_4^i$ and $\mathcal{T}_3^i$ has respectively $l, 2l$ and $4l$ bits. Then, we have

$$\begin{aligned} len(W_i) &= (|\mathcal{T}_5^i| + |\mathcal{T}_4^i| \times r + |\mathcal{T}_3^i| \times r^2)l \\ &= (6 + 15 \times 2 + 20 \times 4)l = 116l, \quad \forall i \in [3]. \end{aligned}$$

Caching: The cache content of user $k \in [2]$ is determined by the key $S_k, k \in [2]$ which is shared only between the server and user k. Shared key $S_k, k \in [2]$ is distributed as $S_k \sim unif\{[N]\} = unif\{[3]\}$. The cache contents of each user is grouped into three parts. The $j^{th}, j = 1, 2, 3$ part of user $k \in [2]$ is denoted by $\mathcal{G}_{k,j}$ and shown in Table I. Thus, the number of bits stored at one user is given by $3\left(\binom{5}{4} + 2 \times \binom{5}{3} + 4 \times \binom{5}{2}\right)l = 195l$. Thus, we have $M = \frac{195}{116}$. Other than S_k the server also places some additional random keys of negligible size in the cache of user $k \in [2]$. These will be used as one-time pads in the delivery phase.

$\mathcal{G}_{k,1}$	$(W_{i,\mathcal{R}} \vert W_{i,\mathcal{R}} \in \mathcal{T}_5^i$ and $S_k + 3k \in \mathcal{R})_{i=0,1,2}$
$\mathcal{G}_{k,2}$	$(W_{i,\mathcal{R}} \vert W_{i,\mathcal{R}} \in \mathcal{T}_4^i$ and $S_k + 3k \in \mathcal{R})_{i=0,1,2}$
$\mathcal{G}_{k,3}$	$(W_{i,\mathcal{R}} \vert W_{i,\mathcal{R}} \in \mathcal{T}_3^i$ and $S_k + 3k \in \mathcal{R})_{i=0,1,2}$

TABLE I: Cache contents of user $k, k = 0, 1$

Delivery: In the delivery phase, for given demands (D_0, D_1), we first construct an expanded demand vector $\bar{d}$ of length 6 such that $\bar{d} = (\bar{d}^{(0)}, \bar{d}^{(1)})$, where $\bar{d}^{(k)}, k = 0, 1$ is obtained by applying $S_k \ominus D_k$ right cyclic shift to the vector $(0, 1, 2)$, where $\ominus$ denotes modulo 3 subtraction. That is, for $k = 0, 1$, $d_i^{(k)} = i - (S_k - D_k) \mod 3$. We now define symbols $Y_{\mathcal{R}}$ for $\mathcal{R} \subset [6]$ and $|\mathcal{R}| = 4, 5, 6$ as follows

$$Y_{\mathcal{R}} = \bigoplus_{u\in\mathcal{R}} W_{d_u,\mathcal{R}\setminus\{u\}}$$

where d_u is the $u+1$-th item in $\bar{d}$.

Symbol $Y_{[6]}$ as defined above is a part of the main payload in the broadcast transmission which needs l bits. To give the other parts of the broadcast, we define symbols $W_{\mathcal{R}}$ and $V_{\mathcal{R}}$ for $\mathcal{R} \subset [6]$ and $|\mathcal{R}| = 4, 5$ as follows

$$W_{\mathcal{R}} = (W_{0,\mathcal{R}} \oplus W_{1,\mathcal{R}}, W_{1,\mathcal{R}} \oplus W_{2,\mathcal{R}}),\ V_{\mathcal{R}} = Y_{\mathcal{R}} \oplus W_{\mathcal{R}}.$$

Note that for $|\mathcal{R}| = 4, W_{\mathcal{R}}$ has two parts, each of length $2l$ bits. $Y_{\mathcal{R}}$ also has a length of $4l$ bits. We further define sets V_4 and V_5 as follows:

$$V_i = \{V_{\mathcal{R}} | \mathcal{R} \cap \{S_0, S_1 + 3\} \neq \phi, |\mathcal{R}| = i\}, \quad \text{for } i = 4, 5.$$

Observe that V_4 and V_5 contain 14 symbols each of size $4l$ and 6 symbols each of size $2l$, respectively. The server picks permutations π_4 and π_5 uniformly at random from respectively the symmetric group of permutations of $[14]$ and $[6]$ and includes $\pi_4(V_4)$ and $\pi_5(V_5)$ in the broadcast. The server does not fully reveal these permutations with any of the users. The position of any symbol $V_{\mathcal{R}} \in V_i, i = 4, 5$ in $\pi_i(V_i)$ is privately transmitted to user k, if and only if $S_k + 3k \in \mathcal{R}$. This private transmission of positions is achieved using one-time pad whose keys are deployed in the caches of respective users in the caching phase. The main payload, X' is given as

$$X' = (X_0, X_1, X_2) = (Y_{[6]}, \pi_4(V_4), \pi_5(V_5)).$$

Thus, the total number of bits transmitted are $(1+6\times2+14\times4)l = 69l$. So, the rate of transmission is $\frac{69}{116}$. Along with the main payload X', the server also broadcasts some auxiliary transmission $J = (S_0 \ominus D_0, S_1 \ominus D_1, J') = (\bar{S} \ominus \bar{D}, J')$. Here, J' contains the positions of various symbols in X_1 and X_2 encoded using one-time pad as discussed above. Thus, the complete broadcast transmission is $X = (X', J)$.

Decoding: For user $k \in [2]$, let us first consider the recovery of segments belonging to $\mathcal{T}_i^{D_k}$, $i = 3, 4$. This is done using symbols from X_1 and X_2. All symbols $W_{D_k,\mathcal{R}} \in \mathcal{T}_i^{D_k}$ where $S_k + 3k \in \mathcal{R}$, i.e., all symbols in set $\mathcal{G}_{k,6-i}$ are cached at user

k. User k decodes the remaining symbols in $\mathcal{T}_i^{D_k}$, i.e., $W_{D_k,\mathcal{R}}$ such that $|\mathcal{R}| = i, S_k + 3k \notin \mathcal{R}$ and $\mathcal{R} \subset [6]$ as follows

$$\widehat{W}_{D_k,\mathcal{R}} = V_{\mathcal{R}^+} \oplus W_{\mathcal{R}^+} \oplus \left(\bigoplus_{u \in \mathcal{R}} W_{d_u,\mathcal{R}^+\setminus\{u\}} \right) \quad (5)$$

where $\mathcal{R}^+ = \{S_k+3k\} \cup \mathcal{R}$. Here, $V_{\mathcal{R}^+}$ is a part of $\pi_{i+1}(V_{i+1})$ and its position in $\pi_{i+1}(V_{i+1})$ has been revealed to user k since $S_k + 3k \in \mathcal{R}^+$. The symbols $W_{\mathcal{R}^+}$ and $W_{d_u,\mathcal{R}^+\setminus\{u\}}$ in (5) can be recovered from her cache. Substituting for $V_{\mathcal{R}^+}$ in (5),

$$\begin{aligned}\widehat{W}_{D_k,\mathcal{R}} &= Y_{\mathcal{R}^+} \oplus W_{\mathcal{R}^+} \oplus W_{\mathcal{R}^+} \oplus \left(\bigoplus_{u \in \mathcal{R}} W_{d_u,\mathcal{R}^+\setminus\{u\}} \right) \\ &= \bigoplus_{u \in \mathcal{R}^+} W_{d_u,\mathcal{R}^+\setminus\{u\}} \oplus \left(\bigoplus_{u \in \mathcal{R}} W_{d_u,\mathcal{R}^+\setminus\{u\}} \right) \\ &= W_{d_{S_k+3k},\mathcal{R}} = W_{D_k,\mathcal{R}}. \end{aligned} \quad (6)$$

Now that user k has all segments in $\mathcal{T}_3^{D_k}$ and $\mathcal{T}_4^{D_k}$, we look at the recovery of symbols in $\mathcal{T}_5^{D_k}$. In the first part $\mathcal{G}_{k,1}$ of cache, user k does not have one segment of $\mathcal{T}_5^{D_k}$, namely $W_{D_k,[6]\setminus\{S_k+3k\}}$. User k decodes this segment as

$$\widehat{W}_{D_k,[6]\setminus\{S_k+3k\}} = Y_{[6]} \oplus \left(\bigoplus_{u \in [6]\setminus\{S_k+3k\}} W_{d_u,[6]\setminus\{u\}} \right).$$

Observe that $Y_{[6]}$ is broadcasted by the server while each symbol $W_{d_u,[6]\setminus\{u\}}$ is a part of $\mathcal{G}_{k,1}$, and hence a part of the cache of user k. Thus, user k can compute $\widehat{W}_{D_k,[6]\setminus\{S_k+3k\}}$. Using ideas from (6), it can be shown that $\widehat{W}_{D_k,[6]\setminus\{S_k+3k\}} = W_{D_k,[6]\setminus\{S_k+3k\}}$(see details in [12]). Thus, user k can retrieve all symbols belonging to each of the three groups of file W_{D_k} and she can recover this file by concatenating these symbols.

Privacy: To show the demand-privacy for user $k \in [2]$, we first define $\tilde{k} = (k+1) \bmod 2$. Since $I(D_{\tilde{k}}; Z_k, D_k) = 0$, the privacy condition $I(D_{\tilde{k}}; X, Z_k, D_k) = 0$ follows by showing that $I(X; D_{\tilde{k}}|Z_k, D_k) = 0$. To that end, we divide all symbols in X' into two sets, X'_k and $\tilde{X}'_k$ which are defined as follows:

$$\begin{aligned} X'_k &= \{Y_{[NK]}\} \cup \{V_{\mathcal{R}} | S_k + 3k \in \mathcal{R}, V_{\mathcal{R}} \in X'\}, \\ \tilde{X}'_k &= X' \setminus X'_k. \end{aligned}$$

Note, that the positions in X' of all symbols belonging to X'_k is known to user k while the positions of symbols belonging to $\tilde{X}'_k$ is not known. It can be shown that all symbols in $\tilde{X}'_k$ appear like a sequence of random bits to user k. This is because for some set $\mathcal{R}$, $\mathcal{R} \subset [6], |\mathcal{R}| = 4, 5$ we broadcast $V_{\mathcal{R}}$ instead of $Y_{\mathcal{R}}$. The symbol $W_{\mathcal{R}}$ essentially hides the message $Y_{\mathcal{R}}$ from all users that do not belong to set $\mathcal{R}$. Further, it can be also shown that

$$H(X'_k | W_{D_k}, Z_k, \bar{S} \ominus \bar{D}) = 0. \quad (7)$$

It is easy to see that, $(W_{D_k}, Z_k, \bar{S} \ominus \bar{D})$ does not reveal any information about $D_{\tilde{k}}$ which in combination with (7) ensures privacy. A rigorous and detailed proof can be found in the extended version [12].

C. Tightness of the achievable memory-rate pairs for $K < N$

We compare the memory-rate pairs achievable using Scheme A with lower bounds on the optimal rates for non-private schemes. Let $R^A_{N,K}(M)$ denote the lower convex envelope of the memory-rate pairs in Theorem 1.

Theorem 2 *The lower convex envelope of memory-rate pairs in Theorem* 1 *and the optimal rates without privacy for $K < N$ always satisfy the following:*

$$\frac{R^A_{N,K}(M)}{R^*_{N,K}(M)} \leq \begin{cases} 3 & \text{if } M < \frac{N}{2} \\ 2 & \text{if } M \geq \frac{N}{2}. \end{cases}$$

Since $R^A_{N,K}(M) \geq R^{*p}_{N,K}(M) \geq R^*_{N,K}(M)$, the same upper bounds also hold for the ratios $\frac{R^A_{N,K}(M)}{R^{*p}_{N,K}(M)}$ and $\frac{R^{*p}_{N,K}(M)}{R^*_{N,K}(M)}$.

The proof of Theorem 2 is given in the extended version [12]. The above result shows that the rates achieved in Theorem 1 are within a multplicative factor of 3 from the optimal for $K < N$. This gives an improvement on the known multiplicative factor of 8 from [4, Theorem 3]. Furthermore, also note that the optimal rates with and without demand privacy are always within a multiplicative factor of 3 for $K < N$.

D. Scheme B

Construction of an (N, K, M, R) demand-private scheme using an (N, NK, M, R) non-private scheme as blackbox has been discussed in [2], [3]. Moreover, it was shown in [4, Theorem 4] that the (N, NK, M, R) non-private scheme needs to serve only a restricted demand type. We construct a novel (N, NK, M, R) non-private scheme which exploits the idea of satisfying only a subset of the general demand set to obtain an improved memory-rate tradeoff for this blackbox. The construction of this scheme is based on the non-private YMA scheme [10] for N files and $NK - K + 1$ users, hence the memory-rate tradeoff achieved by this scheme is same as the $(N, NK - K + 1, M, R)$ YMA scheme. The memory-rate tradeoff achieved by the proposed non-private scheme and thus the corresponding (N, K, M, R) demand-private scheme is given by the following theorem.

Theorem 3 *There exists an (N, K, M, R)-private scheme with the following memory-rate pairs:*

$$(M, R) = \left(\frac{Nr}{NK - K + 1}, \frac{\binom{NK-K+1}{r+1} - \binom{NK-K-N+1}{r+1}}{\binom{NK-K+1}{r}} \right)$$

where $r \in [NK - K + 1]$.

The details of the scheme which achieves the memory-rate pairs in Theorem 3 are delegated to the extended version [12].

As noted before, Scheme B outperforms Scheme A when $N \leq K$. Since the YMA scheme is optimal among all coded caching schemes with uncoded prefetching, the rates achieved by Scheme B will be better than the rates obtained using any (N, NK)-non-private scheme with uncoded prefetching as a blackbox. However, it is not clear whether Scheme B leads to any improvement on the order optimality when $N \leq K$.

IV. Exact Trade-off for $N > 2$ and $K = 2$

For $N = K = 2$, the exact trade-off for private coded caching was characterized in [4]. In the following theorem, we characterize the exact trade-off for $N > 2$ and $K = 2$ under demand privacy.

Theorem 4 *Any memory-rate pair (M, R) is achievable with demand privacy for $N > 2$ and $K = 2$ if and only if*

$$3M + NR \geq 2N, \quad 3M + (N+1)R \geq 2N + 1,$$
$$M + NR \geq N. \tag{8}$$

For $N > 2$ and $K = 2$, any feasible (M, R) pair is required to satisfy the first and third inequalities in (8) even for non-private schemes [11]. The necessity of the second line is shown by generalizing the converse bound given in [4], and the details are delegated to the extended version [12]. The corner points of the memory-rate curve given by (8) are $(0, 2), (\frac{N}{3}, 1), (\frac{N^2}{2N-1}, \frac{N-1}{2N-1})$ and $(N, 0)$. The achievability of $(0, 2)$ and $(N, 0)$ follows from [4, Theorem 2]. We propose two schemes, Scheme C and Scheme D, which achieve memory-rate pairs $(\frac{N}{3}, 1)$ and $(\frac{N^2}{2N-1}, \frac{N-1}{2N-1})$, respectively. Scheme C achieves memory-rate pair $(\frac{N}{3}, 1)$ using uncoded prefetching while Scheme D achieves memory-rate pair $(\frac{N^2}{2N-1}, \frac{N-1}{2N-1})$ using coded prefetching. Next we illustrate Scheme C for $N = 3$, $K = 2$. General versions of Schemes C and D can be found in the extended version [12].

A. An Example of Scheme C for $(N, K, M) = (3, 2, \frac{N}{3})$

We describe Scheme C for $N = 3$ and $K = 2$ which achieves rate 1 for $M = \frac{N}{3} = 1$. File $W_i, i \in [3]$ is divided into 3 disjoint parts of equal size, i.e., $W_i = (W_{i,0}, W_{i,1}, W_{i,2})$.

Caching: The server picks 2 independent permutations π_0 and π_1 uniformly at random from the symmetric group of permutations of [3]. The server places $\pi_0(W_{0,0}, W_{1,0}, W_{2,0})$ and $\pi_1(W_{0,1}, W_{1,1}, W_{2,1})$ in the caches of user 0 and user 1, respectively. Each of these permutation functions π_0 and π_1 are unknown to both the users. Some additional random bits are shared with each user through the cache.

Delivery: The server picks permutation π_2 uniformly at random from the symmetric group of permutations of [3] independent of π_0, π_1. The main payload, X' is given by

$$X' = \begin{cases} \pi_2(W_{D_0,1} \oplus W_{D_1,0}, W_{D_0,2}, W_{D_1,2}) & \text{if } D_0 \neq D_1 \\ \pi_2(W_{D_0,1} \oplus W_{m,0}, W_{D_0,2}, W_{D_1,0} \oplus W_{m,1}) & \\ & \text{if } D_0 = D_1 \end{cases}$$

where $m = (D_0 + 1) \bmod 3$. To enable decoding at each user, the server also transmits some auxiliary transmission $J = (J_1, J_2, J_3)$ of negligible rate. Each $J_j, j = 1, 2, 3$ can be further divided into 2 parts, i.e., $J_j = (J_{j,0}, J_{j,1})$, where $J_{j,k}, k \in [2]$ is meant for user k. Using a one-time pad which uses the pre-shared random bits, the server ensures that $J_{j,k}$ can be decoded only by user k and it is kept secret from the other user. These parts are used as follows:

1) $J_{1,k}$ conveys the position of $W_{D_k,k}$ in user k's cache.
2) $J_{2,k}$ gives the positions of the coded and uncoded parts of X' involving W_{D_k} to user k. Specifically, $J_{2,k}$ reveals the positions of $W_{D_0,1} \oplus W_{D_1,0}$ and $W_{D_k,2}$ to user k when $D_0 \neq D_1$, and the positions of $W_{D_k,\tilde{k}} \oplus W_{m,k}$ and $W_{D_k,2}$ when $D_0 = D_1$, where $\tilde{k} = (k+1) \bmod 2$.
3) $J_{3,k}$ discloses the position of $W_{D_{\tilde{k}},k}$ if $D_0 \neq D_1$ and $W_{m,k}$ if $D_0 = D_1$ in her cache to user k.

Decoding: User k decodes W_{D_k} as follows. $W_{D_k,k}$ can be obtained from the cache since she knows its position from $J_{1,k}$. User k recovers $W_{D_k,2}$ from the delivery since she knows its position in X' from $J_{2,k}$. The remaining segment $W_{D_k,\tilde{k}}$ is available in coded form in X'. The segment that $W_{D_k,\tilde{k}}$ is XOR-ed with, is available in the cache of user k, and its position in the cache is revealed by $J_{3,k}$. Thus, user k retrieves all three segments of file W_{D_k}.

Privacy: Now we describe how D_1 remains private to user 0. From the transmission, we can observe that for both the cases, i.e., $D_0 \neq D_1$ and $D_0 = D_1$, user 0 receives $W_{D_0,2}$ in the uncoded form and $W_{D_0,1}$ coded with another symbol. Also, in both the cases, the remaining symbol is like a sequence of $\frac{F}{3}$ random bits to user 0. This symmetry helps in achieving privacy. Further, given $J_{1,0}$, any of the remaining 2 symbols can occupy the remaining 2 positions in the cache with equal likelihood. Thus, although user 0 can use one of these symbols, i.e., the symbol XOR-ed with $W_{D_0,1}$, for decoding using $J_{3,0}$, the symbol's identity is unknown because $J_{3,0}$ only discloses the symbol's position in user 0's cache. Due to the symmetry of the scheme, similar privacy arguments apply for user 1.

References

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

[2] K. Wan and G. Caire, "On coded caching with private demands," arXiv:1908.10821 [cs.IT], Jun. 2020.

[3] S. Kamath, "Demand private coded caching," arXiv:1909.03324 [cs.IT], Sep. 2019.

[4] S. Kamath, J. Ravi, and B. K. Dey, "Demand-private coded caching and the exact trade-off for N=K=2," in *2020 National Conference on Communications (NCC)*, Kharagpur, India, Feb. 2020.

[5] V. R. Aravind, P. K. Sarvepalli, and A. Thangaraj, "Subpacketization in coded caching with demand privacy," in *2020 National Conference on Communications (NCC)*, Kharagpur, India, Feb. 2020.

[6] Q. Yan and D. Tuninetti, "Fundamental limits of caching for demand privacy against colluding users," arXiv:2008.03642 [cs.IT], Aug. 2020.

[7] K. Wan, H. Sun, M. Ji, D. Tuninetti, and G. Caire, "Fundamental limits of device-to-device private caching with trusted server," arXiv:1912.09985 [cs:IT], Jan. 2020.

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

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

[10] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *IEEE Transactions on Information Theory*, vol. 64, no. 2, pp. 1281–1296, Feb. 2018.

[11] C. Tian, "Symmetry, outer bounds, and code constructions: A computer-aided investigation on the fundamental limits of caching," *Entropy*, vol. 20, no. 8, pp. 603.1–603.43, Aug. 2018.

[12] C. Gurjarpadhye, J. Ravi, S. Kamath, B. K. Dey, and N. Karamchandani, "Fundamental limits of demand-private coded caching," arXiv:2101.07127 [cs.IT], Jan. 2021.