Device-to-Device Private Caching with Trusted Server

Kai Wan*, Hua Sun†, Mingyue Ji‡, Daniela Tuninetti§, Giuseppe Caire*
*Technische Universität Berlin, 10587 Berlin, Germany, {kai.wan, caire}@tu-berlin.de
†University of North Texas, Denton, TX 76203, USA, hua.sun@unt.edu
‡University of Utah, Salt Lake City, UT 84112, USA, mingyue.ji@utah.edu
§University of Illinois at Chicago, Chicago, IL 60607, USA, danielat@uic.edu

***Abstract*—In order to preserve the privacy of the users demands from other users, in this paper we formulate a novel information theoretic Device-to-Device (D2D) private caching model by adding a trusted server. In the delivery phase, the trusted server collects the users demands and sends a query to each user, who then broadcasts packets according to this query. Two D2D private caching schemes (uncoded and coded) are proposed in this paper, which are shown to be order optimal.**

I. Introduction

Coded caching was originally proposed by Maddah-Ali and Niesen (MAN) for shared-link networks [1], where a server with access to a library of N files is connected to K users through an error-free broadcast link. Each user can store up to M files at its cache. The MAN caching scheme includes *placement* and *delivery* phases. In the placement phase without knowing the later demands, letting $t = \mathsf{KM}/\mathsf{N} \in [0 : \mathsf{K}]$ represent the ratio between the size of the aggregate cache memory of the K users and the library size, each file is divided into $\binom{\mathsf{K}}{t}$ subfiles, each of which is cached by a different t-subset of users. In the delivery phase, each user demands one file. According to the users demands, the server sends $\binom{\mathsf{K}}{t+1}$ MAN multicast messages, each of which is useful to $t+1$ users simultaneously (i.e., the coded caching/multicasting gain is $t+1$). It was proved in [2] that the worst-case load achieved by the MAN scheme among all possible demands is optimal under the constraint of uncoded placement (i.e., each user directly stores packets from the library files, rather than more general functions thereof) and $\mathsf{N} \geq \mathsf{K}$. When $\mathsf{N} \geq \mathsf{K}$, the MAN scheme was also proved in [3] to be generally order optimal within a factor of 2. By observing that some MAN multicast messages when $\mathsf{N} < \mathsf{K}$ are redundant, an improved delivery scheme was proposed in [4], which was proved to be optimal under the constraint of uncoded cache placement, and order optimal within a factor of 2 without any constraint on the placement; we shall refer to such a scheme as YMA delivery.

Coded caching strategy was then extended to Device-to-Device networks by Ji, Caire, and Molisch (JCM) [5], where in the delivery phase each user broadcasts packets in function of its cached content and the users demands, to all other users. With the MAN cache placement, JCM splits each MAN multicast message into $t+1$ equal-length sub-messages, each of which is conveyed to the other users by a user with the MAN delivery. By replacing the MAN delivery with the YMA delivery, the scheme (which effectively splits the D2D network into K parallel shared-link models) is order optimal to within a factor of 4 as proved by Yapar et *al.* (YWSC) [6].

For the successful decoding of a MAN multicast message, users need to know the composition of this message (i.e., which subfiles are coded together). As a consequence, users are aware of the demands of other users, which is problematic in terms of privacy. Shared-link coded caching with private demands, which aims to preserve the privacy of the users' demands from other users, was originally discussed in [7] and recently analyzed information-theoretically by Wan and Caire (WC) in [8], where two schemes were proposed and shown to be order optimal. Relevant to this paper is the second scheme in [8] also discussed in [9], which operates as if there are KN users in total, i.e., it pretends there are $\mathsf{NK}-\mathsf{K}$ virtual users in addition to the K real users, so that each file is demanded exactly K times. Such a scheme is order optimal to within a factor of 8. By observing that the private caching schemes in [8], [9] need high subpacketiation levels (i.e., the number of subfiles into which each file must be partitioned in the placement phase), the authors in [10] proposed a private caching scheme for two-user and two-file systems, with the minimal possible subpacketization level.

In this paper, we consider the problem of coded caching with private demands for D2D systems. We introduce a novel D2D architecture with a *trusted server*. This trusted server is connected to each user through an individual link and without access to the library, as illustrated in Fig. 1. The placement phase is the same as the shared-link and D2D caching models. In the delivery phase, each user first informs the trusted server about the index of the demanded file. After collecting the information about the users' demands and the cached content, the trusted server sends a query to each user. Given the query, each user then broadcasts packets accordingly. *The trusted server acts only as a coordinator to warrant demand privacy, but does not support any large load of communication.* The demands and the control commands to tell the users what to send can be seen as protocol information, requiring a communication load negligible with respect to the actual file transmission. Hence, the load of the system is still only supported by D2D communication, while the user-server communication is only protocol information. The objective is to design a two-phase private D2D caching scheme

978-1-7281-5089-5/20/$31.00 ©2020 IEEE

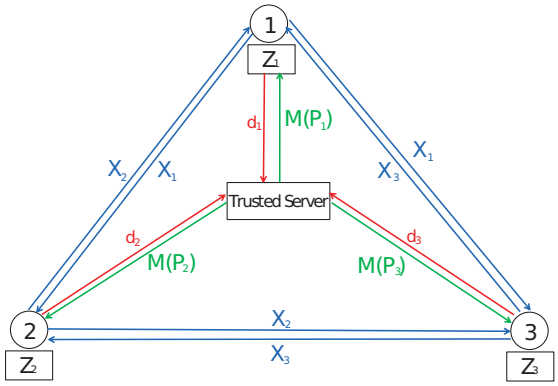

Fig. 1: The formulated D2D private caching problem with a trusted server and $\mathsf{K} = 3$ users.

with minimum number of transmitted bits by all users in the delivery phase, while preserving the users demands from the other users.

The main contributions in this paper are as follows: (i) We give an information-theoretic formulation of the D2D coded caching problem with demand privacy. (ii) We propose two schemes. A baseline uncoded scheme that essentially delivers the whole library to all users, which is trivially private, and a coded scheme that carefully combines the idea of introducing virtual users as in [8], [9] with that a splitting the D2D network into multiple shared link ones as in [5], [6]. (iii) By comparing with the converse bound for the shared-link caching problem without privacy constraint in [3], we prove the proposed coded scheme is order optimal to within a factor of 6 when $\mathsf{N} \geq \mathsf{K}$ and $\mathsf{M} \geq 2\mathsf{N}/\mathsf{K}$, and within a factor of 12 when $\mathsf{N} < \mathsf{K}$ and $\mathsf{M} \geq \mathsf{N}/\mathsf{K}$.

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; we let $\binom{x}{y} = 0$ if $x < 0$ or $y < 0$ or $x < y$.

II. System Model and Related Results

A. System Model

A $(\mathsf{K}, \mathsf{N}, \mathsf{M})$ D2D private caching system with a trusted server is defined as follows. The library contains N independently generated files, denoted by $(F_1, F_2, \ldots, F_{\mathsf{N}})$, where each file is composed of B i.i.d. bits, where B is assumed sufficiently large such that any sub-packetization of the files is possible. There are K users in the system, each of which is equipped with a cache of MB bits, where $\mathsf{M} \in \left[\frac{\mathsf{N}}{\mathsf{K}}, \mathsf{N}\right]$. There is a trusted server without access to the library in the system. This server is connected to each user through an individual secure link. In addition, there is also a broadcast link from each user to all other users (e.g., a shared medium)[1]. We only consider the case $\min\{\mathsf{K}, \mathsf{N}\} \geq 2$, since when $\mathsf{K} = 1$ or $\mathsf{N} = 1$ each user knows the demand of other users.

Let $\epsilon_{\mathsf{B}} \geq 0$ be a constant. The system operates in two phases.

Placement Phase. Each user $k \in [\mathsf{K}]$ stores content in its cache without knowledge of later demand. We denote the content in the cache of user $k \in [\mathsf{K}]$ by

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

where C_k represents the cached content, a function of the N files, and $\mathscr{M}(C_k)$ represents the metadata/composition of C_k (i.e., how C_k is generated). We have

$$H\big(C_k|\mathscr{M}(C_k), F_1, \ldots, F_N\big) = 0 \text{ (placement constraint)}, \tag{2}$$

i.e., C_k is a deterministic function of the library and of the metadata describing the cache encoding. 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 the trusted server, and is not known by other users. The cache content of user $k \in [\mathsf{K}]$ in (1) is constrained by the cache size as

$$H(Z_k) \leq \mathsf{B}(\mathsf{M} + \epsilon_{\mathsf{B}}) \text{ (cache size constraint)}. \tag{3}$$

Delivery Phase. During the delivery phase, each user $k \in [\mathsf{K}]$ demands the file with index d_k, where d_k is a realization of the random variable D_k with range in $[\mathsf{N}]$. The demand vector of the K users, denoted by $\mathbf{D} = (D_1, \ldots, D_{\mathsf{K}})$. The delivery phase contains the following steps:

- Step 1: each user $k \in [\mathsf{K}]$ sends the index of its demanded file (i.e., d_k) to the trusted server.
- Step 2: according to the users' demands and the cache contents, the trusted server where the metadata $\mathscr{M}(P_k)$ describes how the packets P_k, to be broadcasted by user $k \in [\mathsf{K}]$, are composed.
- Step 3: each user $k \in [\mathsf{K}]$ broadcasts $X_k = (\mathscr{M}(P_k), P_k)$ to other users based only on the its local storage content Z_k and the metadata $\mathscr{M}(P_k)$, that is

$$H(X_k|\mathscr{M}(P_k), Z_k) = 0 \text{ (encoding constraint)}. \tag{4}$$

Decoding. Let $\mathbf{X} := (X_j : j \in [\mathsf{K}])$ be the vector of all transmitted signals. To guarantee successful decoding at user $k \in [\mathsf{K}]$ it must hold that

$$H\big(F_{D_k}|\mathbf{X}, Z_k, D_k\big) \leq \mathsf{B}\epsilon_{\mathsf{B}} \text{ (decoding constraint)}, \tag{5}$$

and to guarantee privacy it must hold

$$I\big(\mathbf{D}; \mathbf{X}, Z_k|D_k\big) = 0 \text{ (privacy constraint)}. \tag{6}$$

The privacy constraint in (6) (i.e., vanishing information leakage) corresponds to perfect secrecy in an information theoretic sense (see [11, Chapter 22]).

[1]We assume a collision avoidance protocol for which when a user broadcasts, all the others stay quiet and listen (e.g., this can be implemented in a practical wireless network using CSMA, as in the IEEE 802.11 standard).

Objective. We say that load R is achievable if

$$\sum_{k\in[\mathsf{K}]} H(X_k) \le \mathsf{B}(\mathsf{R}+\epsilon_{\mathsf{B}}) \text{ (load)}, \tag{7}$$

while all the above constraints are satisfied and $\lim_{\mathsf{B}\to\infty}\epsilon_{\mathsf{B}} = 0$. The objective is to determine, for a fixed $\mathsf{M}\in\left[\frac{\mathsf{N}}{\mathsf{K}},\mathsf{N}\right]$, the minimum achievable load, which is indicated by $\mathsf{R}^\star$.

B. Shared-link Private Caching Scheme in [8]

We then recall in short the shared-link private caching scheme proposed in [8, Remark 1] for general demand and memory size regime, whose key strategy in [8] is to generate $\mathsf{NK}-\mathsf{K}$ virtual users such that the system contains NK effective users (i.e., real or virtual users).

Placement Phase. A private placement precoding strategy was proposed in [8], which can concatenate with any uncoded cache placement and any MDS-code based placement.

Now we use this private placement precoding with the MAN cache placement for NK users. More precisely, let $\mathsf{M} = \mathsf{N}t/(\mathsf{NK}) = t/\mathsf{K}$ where $t\in[0:\mathsf{NK}]$. Each file F_i where $i\in[\mathsf{N}]$ is divided into $\binom{\mathsf{NK}}{t}$ non-overlapping and equal-length pieces. For each file F_i, by randomly generating a permutation of $\left[\binom{\mathsf{NK}}{t}\right]$, we assign each piece to one subfile $F_{i,\mathcal{W}}$, where $\mathcal{W}\subseteq[\mathsf{NK}]$ and $|\mathcal{W}|=t$, according to this permutation. Each user $k\in[\mathsf{K}]$ caches $F_{i,\mathcal{W}}$ where $k\in\mathcal{W}$. The random permutation is unknown by each user $k\in[\mathsf{K}]$. As a result, from the viewpoint of user $k\in[\mathsf{K}]$, each cached subfile of file F_i where $i\in[\mathsf{N}]$ is equivalent from the viewpoint of user k, while each uncached subfile of F_i is also equivalent.

Delivery Phase. When the demand vector of the K real users is revealed to the server, the demands of the virtual users are generated such that each file is demanded by exactly K effective users. For each $\mathcal{S}\subseteq[\mathsf{NK}]$ where $|\mathcal{S}|=t+1$, the server generates a MAN multicast message

$$W_{\mathcal{S}} = \underset{k\in\mathcal{S}}{\oplus} F_{d_k,\mathcal{S}\setminus\{k\}}. \tag{8}$$

Then the server, by generating a random permutation of $\left[\binom{\mathsf{NK}}{t+1}\right]$, transmits all $\binom{\mathsf{NK}}{t+1}$ MAN multicast messages in an order according to the random permutation, which is unknown by each user, such that each user does not know the $t+1$ effective users for which each MAN multicast message is useful.

As a result, from the viewpoint of each real user $k\in[\mathsf{K}]$, the compositions of the received multicast messages are equivalent for different demand vectors given d_k, such that it cannot get any information about the demands of other real users.

The JCM caching scheme in [5] extends the K-user MAN caching scheme to the D2D scenario (without privacy constraint) by using the MAN cache placement and splitting each MAN multicast message in the delivery phase into $t+1$ equal-length sub-messages, each of which can be transmitted by one of the $t+1$ users. However, it is difficult to use this extension idea to directly extend the above shared-link private caching to the considered D2D private scenario. The main issue is that we cannot have any virtual transmitter in the system. Hence, instead of directly extending the shared-link private caching scheme proposed in [8] to the D2D private model, we will propose a novel and non-trivial D2D private caching scheme.

III. D2D Caching Schemes with Private Demands

A trivial solution is to let each user recover the whole library in order to hide its demanded file.

Theorem 1 (Uncoded Scheme)**.** *For the* $(\mathsf{K},\mathsf{N},\mathsf{M})$ *private D2D caching system,* $\mathsf{R}^\star$ *is upper bounded by*

$$\mathsf{R}^\star \le \mathsf{R}_u = \frac{\mathsf{K}}{\mathsf{K}-1}(\mathsf{N}-\mathsf{M}). \tag{9}$$

We then propose a coded private caching scheme with a novel cache placement based on generating $(\mathsf{K}-1)(\mathsf{N}-1)$ virtual users whose subpacketization is different from the MAN cache placement, and a novel coded delivery scheme, the compositions of whose transmitted multicast messages are equivalent from the viewpoint of each real user. More precisely, from the novel caching construction, the proposed D2D private caching scheme divides the D2D scenario into K independent shared-link caching models, each of which serves

$$\mathsf{U} := (\mathsf{K}-1)\mathsf{N} \tag{10}$$

effective users. In addition, instead of assigning one demand to each virtual user in the D2D scenario, we assign one demand to each virtual user for each of the K divided shared-link models, such that each file is demanded by $\mathsf{K}-1$ effective users to be served in this shared-link model. The achieved load is given in the following theorem and the detailed description on the proposed scheme could be found in Section III-B.

Theorem 2 (Coded Scheme)**.** *For the* $(\mathsf{K},\mathsf{N},\mathsf{M})$ *D2D private caching system,* $\mathsf{R}^\star$ *is upper bounded by the lower convex envelope of the following points*

$$(\mathsf{M},\mathsf{R}_c) = \left(\frac{\mathsf{N}+t-1}{\mathsf{K}}, \frac{\binom{\mathsf{U}}{t}-\binom{\mathsf{U}-\mathsf{N}}{t}}{\binom{\mathsf{U}}{t-1}}\right),\ \forall t\in[\mathsf{U}+1]. \tag{11}$$

Notice that when $t=\mathsf{U}+1$ in (11), we have the trivial corner point $(\mathsf{M},\mathsf{R}_c)=(\mathsf{N},0)$.

By comparing the proposed coded private caching scheme in Theorem 2 and the converse bound for the shared-link caching problem without privacy constraint in [3], we have the following order optimality results (whose detailed proof could be found in the extended version of this paper [12]).

Theorem 3. *For the* $(\mathsf{K},\mathsf{N},\mathsf{M})$ *private D2D caching system, the proposed scheme in Theorem 2 is order optimal within a factor of* 6 *if* $\mathsf{N}\ge\mathsf{K}$ *and* $\mathsf{M}\ge 2\mathsf{N}/\mathsf{K}$*, and a factor of* 12 *if* $\mathsf{N}<\mathsf{K}$ *and* $\mathsf{M}\ge\mathsf{N}/\mathsf{K}$.

Remark 1. *We say that the users in the system collude if they exchange the indices of their demanded files and their cache contents. Privacy constraint against colluding users is a stronger notion than* (6) *and is defined as follows*

$$I\big(\mathbf{D};\mathbf{X},\{Z_k:k\in\mathcal{S}\}|\{D_k:k\in\mathcal{S}\}\big)=0,\ \forall\mathcal{S}\subseteq[\mathsf{K}],\mathcal{S}\neq\emptyset. \tag{12}$$

In the extended version of this paper [12], we also propose a novel converse bound by considering the privacy constraint in (12)*. Comparing the proposed achievable scheme in Theorem 2 and the novel converse bound, we can prove that the scheme in Theorem 2 is order optimal under the constraint of uncoded cache placement and privacy against colluding users, within a factor of* 18 *(numerical simulations suggest* $27/2$*).*

A. Example

Before the general description on the proposed scheme in Theorem 2, we first use the following example to illustrate the main idea. Consider the $(\mathsf{K},\mathsf{N},\mathsf{M}) = (2,3,2)$ D2D caching system with private demands. From (11) and (10), in this example we have $t = 2$ and $\mathsf{U} = 3$.

Placement Phase. We also use the private placement precoding strategy proposed in [8]. Each file F_i where $i \in [\mathsf{N}]$ is divided into $\mathsf{K}\binom{\mathsf{U}}{t-1} = 6$ non-overlapping and equal-length pieces, denoted by $S_{i,1},\ldots,S_{i,6}$, where each piece has $\mathsf{B}/6$ bits. For user $k_1 = 1$, we aim to generate the subfiles for the shared-link model, in which user $k_1 = 1$ broadcast packets and there are $\mathsf{K}-1 = 1$ real user (user 2) and $(\mathsf{K}-1)(\mathsf{N}-1) = 2$ virtual users (users 3 and 4) to be served. In other words, there are totally $(\mathsf{K}-1)(\mathsf{N}-1)+\mathsf{K}-1 = \mathsf{U}$ effective users to be served, whose union set is $[(\mathsf{K}-1)(\mathsf{N}-1)+\mathsf{K}]\setminus\{k_1\} = [2:4]$. We randomly generate a permutation of $\left[\binom{\mathsf{U}}{t-1}\right] = [3]$, denoted by $\mathbf{p}_{i,k_1} = \mathbf{p}_{i,1} = (p_{i,1}[1],p_{i,1}[2],p_{i,1}[3])$, independently and uniformly over the set of all possible permutations. We assume that $\mathbf{p}_{i,1} = (1,2,3)$. For each set $\mathcal{W} \subseteq [(\mathsf{K}-1)(\mathsf{N}-1)+\mathsf{K}]\setminus\{k_1\} = [2:4]$ where $|\mathcal{W}| = t-1 = 1$, we generate a subfile $f^{k_1}_{i,\mathcal{W}}$ of F_i which should be cached by users in $\{k_1\}\cup\mathcal{W}\cap[\mathsf{K}]$ according to $\mathbf{p}_{i,1}$ as follows,

$$f^1_{i,\{2\}} = S_{i,p_{i,1}[1]} = S_{i,1},\ f^1_{i,\{3\}} = S_{i,p_{i,1}[2]} = S_{i,2}, \quad (13a)$$
$$f^1_{i,\{4\}} = S_{i,p_{i,1}[3]} = S_{i,3}. \quad (13b)$$

Hence, $f^1_{i,\{2\}}$ is cached by users 1 and 2, while $f^1_{i,\{3\}}$ and $f^1_{i,\{4\}}$ are only cached by user 1. Similarly, for user $k_2 = 2$, we randomly generate a permutation of $\left[\binom{\mathsf{U}}{t-1}+1:2\binom{\mathsf{U}}{t-1}\right] = [4:6]$, denoted by $\mathbf{p}_{i,k_2} = \mathbf{p}_{i,2} = (p_{i,2}[1],p_{i,2}[2],p_{i,2}[3])$, independently and uniformly over the set of all possible permutations. We assume that $\mathbf{p}_{i,2} = (4,5,6)$. For each set $\mathcal{W} \subseteq [(\mathsf{K}-1)(\mathsf{N}-1)+\mathsf{K}]\setminus\{k_2\} = \{1,3,4\}$ where $|\mathcal{W}| = t-1 = 1$, we generate a subfile $f^{k_2}_{i,\mathcal{W}}$ of F_i which should be cached by users in $\{k_2\}\cup\mathcal{W}\cap[\mathsf{K}]$ according to $\mathbf{p}_{i,2}$ as follows,

$$f^2_{i,\{1\}} = S_{i,p_{i,2}[1]} = S_{i,4},\ f^2_{i,\{3\}} = S_{i,p_{i,2}[2]} = S_{i,5}, \quad (14a)$$
$$f^2_{i,\{4\}} = S_{i,p_{i,2}[3]} = S_{i,6}. \quad (14b)$$

Hence, $f^2_{i,\{1\}}$ is cached by users 1 and 2, while $f^2_{i,\{3\}}$ and $f^2_{i,\{4\}}$ are only cached by user 2.

Recall that $Z_1 = (\mathscr{M}(C_1),C_1)$ denotes the cache of user 1. In this example,

$$C_1 = \cup_{i\in[\mathsf{N}]}\{S_{i,p_{i,1}[1]},S_{i,p_{i,1}[2]},S_{i,p_{i,1}[3]},S_{i,p_{i,2}[1]}\}$$

and $\mathscr{M}(C_1)$ denotes the indices of the contained bits in C_1.[2] Similarly, we can obtain Z_2 for user 2. Hence, each user $k \in [2]$ caches 4 pieces of each file, and thus it totally caches $2\frac{4\mathsf{B}}{6} = 2\mathsf{B}$ bits, satisfying the memory size constraint. In addition, since the random permutations $\mathbf{p}_{i,1}$ and $\mathbf{p}_{i,2}$ are unknown to user k, each cached subfile of F_i with the same superscript is equivalent from the viewpoint of user k, e,g, $f^1_{i,\{2\}}$, $f^1_{i,\{3\}}$, and $f^1_{i,\{4\}}$ are equivalent from the viewpoint of user 1. Each uncached subfile of F_i with the same superscript is also equivalent from the viewpoint of user k, e.g., $f^2_{i,\{3\}}$ and $f^2_{i,\{4\}}$ are equivalent from the viewpoint of user 1.

We will consider two demand vectors $(1,2)$, $(1,1)$, which represent all possible non-equivalent demand configurations.

Delivery Phase for $\mathbf{d} = (1,2)$. We treat the K transmissions from the K users as K shared-link transmissions.

Let us first consider the 1st shared-link transmission in which user $k_1 = 1$ broadcast packets. We assign one demanded file to each virtual user such that each file in the library is demanded by $\mathsf{K}-1 = 1$ effective user in $[2:4]$. More precisely, we first let $d^1_2 = d_2 = 2$, representing the demanded file by real user 2 in the 1st shared-link transmission. We also let $d^1_3 = 1$ and $d^1_4 = 3$, representing the demanded files by virtual users 3 and 4 in the 1st shared-link transmission, respectively. For each set $\mathcal{S} \subseteq [(\mathsf{K}-1)(\mathsf{N}-1)+\mathsf{K}]\setminus\{k_1\} = [2:4]$ where $|\mathcal{S}| = t = 2$, we generate

$$W^{k_1}_{\mathcal{S}} = \underset{j\in\mathcal{S}}{\oplus} f^{k_1}_{d^{k_1}_j,\mathcal{S}\setminus\{j\}}. \quad (15)$$

In this example, we have

$$W^1_{\{2,3\}} = f^1_{2,\{3\}}\oplus f^1_{1,\{2\}},\quad W^1_{\{2,4\}} = f^1_{2,\{4\}}\oplus f^1_{3,\{2\}}, \quad (16a)$$
$$W^1_{\{3,4\}} = f^1_{1,\{4\}}\oplus f^1_{3,\{3\}}. \quad (16b)$$

Finally, we generate one permutation of $\left[\binom{\mathsf{U}}{t}\right] = [3]$, denoted by $\mathbf{q}_{k_1} = \mathbf{q}_1 = (q_{1,1},q_{1,2},q_{1,3})$, independently and uniformly over the set of all possible permutations. By assuming $\mathbf{q}_1 = (1,2,3)$ which is used to transmit the three multicast messages in (16) in an order which is unknown to users, we can hide the users to whom each multicast message is useful. Hence, we let $P_{k_1} = P_1 = (W^1_{\{2,3\}},W^1_{\{2,4\}},W^1_{\{3,4\}})$. The trusted server transmits $\mathscr{M}(P_1)$ to user 1, who is then instructed to broadcast $X_1 = (\mathscr{M}(P_1),P_1)$.

We then consider the 2nd shared-link transmission, in which user $k_2 = 2$ broadcast packets as the server.Similarly, we let $d^2_1 = d_1 = 1$, $d^2_3 = 2$, $d^2_4 = 3$, and generate

$$W^2_{\{1,3\}} = f^2_{1,\{3\}}\oplus f^2_{2,\{1\}},\ W^2_{\{1,4\}} = f^2_{1,\{4\}}\oplus f^2_{3,\{1\}}, \quad (17a)$$
$$W^2_{\{3,4\}} = f^2_{2,\{4\}}\oplus f^2_{3,\{3\}}. \quad (17b)$$

By generating a random permutation of [3], denoted by $\mathbf{q}_2$ (assumed to be $(1,2,3)$), we let $P_2 = (W^2_{\{1,3\}},W^2_{\{1,4\}},W^2_{\{3,4\}})$. The trusted server transmits $\mathscr{M}(P_2)$ to user 2, who is then instructed to broadcast $X_2 = (\mathscr{M}(P_2),P_2)$. From the received

[2] For sake of simplicity, in the rest of paper, when we describe our achievable scheme, we directly provide C_k or X_k for each user $k \in [\mathsf{K}]$ without repeating that its metadata.

packets, each user $k \in [2]$ can recover its demanded file. For the privacy, we then focus on the demand vector $\mathbf{d} = (1, 1)$.

Delivery Phase for $\mathbf{d} = (1, 1)$. By the same method as described above, user 1 is instructed to broadcast $X_1 = (\mathscr{M}(P_1), P_1)$ where $P_1 = (W^1_{\{2,3\}}, W^1_{\{3,4\}}, W^1_{\{2,4\}})$[3] and

$$W^1_{\{2,3\}} = f^1_{1,\{3\}} \oplus f^1_{2,\{2\}}, \quad W^1_{\{3,4\}} = f^1_{2,\{4\}} \oplus f^1_{3,\{3\}}, \tag{18a}$$
$$W^1_{\{2,4\}} = f^1_{1,\{4\}} \oplus f^1_{3,\{2\}}. \tag{18b}$$

User 2 is instructed to broadcast $X_2 = (\mathscr{M}(P_2), P_2)$ where $P_2 = (W^2_{\{1,3\}}, W^2_{\{1,4\}}, W^2_{\{3,4\}})$ and

$$W^2_{\{1,3\}} = f^2_{1,\{3\}} \oplus f^2_{2,\{1\}}, \quad W^2_{\{1,4\}} = f^2_{1,\{4\}} \oplus f^2_{3,\{1\}}, \tag{19a}$$
$$W^2_{\{3,4\}} = f^2_{2,\{4\}} \oplus f^2_{3,\{3\}}. \tag{19b}$$

Privacy. Let us focus on user 1. For each demand vector, the delivery scheme is equivalent to two independent shared-link transmissions, and in the k^{th} shared-link transmission where $k \in [2]$ only the subfiles with superscript k are transmitted by user k. In other words, no subfile appears in the two shared-link transmissions simultaneously. Each shared-link transmission is equivalent to a shared-link private caching scheme in [8] where each file in the library is demanded by $\mathsf{K} - 1 = 1$ effective users. By the construction on the cache placement, each cached subfile of F_i with the same superscript is equivalent from the viewpoint of each real user, while each uncached subfile of F_i with the same superscript is also equivalent from the viewpoint of this user. Hence, the k^{th} shared-link transmissions for different demand vectors are equivalent from the viewpoint of each real user. For example, for user 1, $f^1_{1,\{2\}}$ and $f^1_{1,\{3\}}$ are equivalent, while $f^1_{2,\{3\}}$ and $f^1_{2,\{2\}}$ are equivalent. Hence, $f^1_{2,\{3\}} \oplus f^1_{1,\{2\}}$ transmitted for demand $(1, 2)$ is equivalent to $f^1_{1,\{3\}} \oplus f^1_{2,\{2\}}$ transmitted for demand $(1, 1)$ from the viewpoint of user 1. Similarly, X_1 transmitted for demand $(1, 2)$ is equivalent to X_1 transmitted for demand $(1, 1)$ from the viewpoint of user 1. By the same reasoning, it can be checked that X_2's transmitted for different demands are also equivalent from the viewpoint of user 1. In conclusion, user 1 does not know any information about the demand of user 2 from the transmission.

Similarly, it can be seen that the privacy of the demand of user 1 is also preserved from user 2. Hence, the proposed D2D coded private caching scheme is indeed private.

Performance. Each user transmits three binary sums of subfiles, each of which has $\mathsf{B}/6$ bits. Hence, the achieved load is 1, while the load achieved by the uncoded scheme in Theorem 1 is 2 and the JCM caching scheme without privacy achieves $2/3$.

B. Proof of Theorem 2

We are now ready to generalize the example in Section III-A. Recall $\mathsf{U} = (\mathsf{K} - 1)\mathsf{N}$ defined in (10). We focus on the memory size $\frac{(\mathsf{K}-1)(t-1)+\mathsf{U}}{\mathsf{KU}}\mathsf{N}$, where $t \in [\mathsf{U}]$. We generate $(\mathsf{K} - 1)(\mathsf{N} - 1)$ virtual users, which are labelled as users $\mathsf{K} + 1, \ldots, (\mathsf{K} - 1)(\mathsf{N} - 1) + \mathsf{K}$.

Placement Phase. Each file F_i where $i \in [\mathsf{N}]$ is divided into $\mathsf{K}\binom{\mathsf{U}}{t-1}$ non-overlapping and equal-length pieces, denoted by $S_{i,1}, \ldots, S_{i,\mathsf{K}\binom{\mathsf{U}}{t-1}}$, where each piece has $\frac{\mathsf{B}}{\mathsf{K}\binom{\mathsf{U}}{t-1}}$ bits. For each user $k \in [\mathsf{K}]$, we aim to generate the subfiles for the k^{th} shared-link model, in which user k broadcast packets as the server and there are $\mathsf{K} - 1 = 1$ real user and $(\mathsf{K} - 1)(\mathsf{N} - 1) = 2$ virtual users to be served. In other words, there are totally $(\mathsf{K} - 1)(\mathsf{N} - 1) + \mathsf{K} - 1 = \mathsf{U}$ effective users to be served, whose union set is $[(\mathsf{K} - 1)(\mathsf{N} - 1) + \mathsf{K}] \setminus \{k\}$. We randomly generate a permutation of $\left[(k-1)\binom{\mathsf{U}}{t-1} + 1 : k\binom{\mathsf{U}}{t-1}\right]$, denoted by $\mathbf{p}_{i,k} = \left(p_{i,k}[1], \ldots, p_{i,k}\left[\binom{\mathsf{U}}{t-1}\right]\right)$, independently and uniformly over the set of all possible permutations. We sort all sets $\mathcal{W} \subseteq [(\mathsf{K}-1)(\mathsf{N}-1)+\mathsf{K}]\setminus\{k\}$ where $|\mathcal{W}| = t-1$, in a lexicographic order, denoted by $\mathcal{W}(1), \ldots, \mathcal{W}\left(\binom{\mathsf{U}}{t-1}\right)$. For each $j \in \left[\binom{\mathsf{U}}{t-1}\right]$, we generate a subfile

$$f^k_{i,\mathcal{W}(j)} = S_{i,p_{i,k}[j]}, \tag{20}$$

which is cached by users in $\{k\} \cup \mathcal{W}(j) \cap [\mathsf{K}]$.

After considering all K shared-link models, each real user $k \in [\mathsf{K}]$ caches all $\binom{\mathsf{U}}{t-1}$ subfiles with superscript k, and $\binom{\mathsf{U}-1}{t-2}$ subfiles with superscript k' for each $k' \in [\mathsf{K}] \setminus \{k\}$. Hence, user k totally caches $\binom{\mathsf{U}}{t-1} + (\mathsf{K} - 1)\binom{\mathsf{U}-1}{t-2}$ subfiles, each of which has $\frac{\mathsf{B}}{\mathsf{K}\binom{\mathsf{U}}{t-1}}$ bits, and thus the number of cached bits is $\frac{\binom{\mathsf{U}}{t-1}+(\mathsf{K}-1)\binom{\mathsf{U}-1}{t-2}}{\mathsf{K}\binom{\mathsf{U}}{t-1}}\mathsf{B} = \mathsf{MB}$. Moreover, for each file $i \in [\mathsf{N}]$, the random permutations $\mathbf{p}_{i,j}$ where $j \in [\mathsf{K}]$ are unknown to user $k \in [\mathsf{K}]$. Hence, from the viewpoint of user k, each cached subfile of F_i with the same superscript is equivalent from the viewpoint of user k, while each uncached subfile of F_i with the same superscript is also equivalent.

Delivery Phase. We divide the transmissions from the K into K shared-link transmissions. Let us focus on the k^{th} shared-link transmission, where $k \in [\mathsf{K}]$.

We first assign one demanded file to each virtual user such that each file in the library is demanded by $\mathsf{K} - 1$ effective user. More precisely, for each real user $k' \in [\mathsf{K}] \setminus \{k\}$, let

$$d^k_{k'} = d_{k'}. \tag{21}$$

We then define

$$n_{i,k} := |\{k' \in [\mathsf{K}] \setminus \{k\} : d_{k'} = i\}|, \ \forall i \in [\mathsf{N}], \tag{22}$$

which represents the number of real users in $[\mathsf{K}]\setminus\{k\}$ demanding F_i. One file is assigned to each of the $(\mathsf{K}-1)(\mathsf{N}-1)$ virtual users as follows. For each file $i \in [\mathsf{N}]$, we let

$$d^k_{1+\mathsf{K}+(i-1)(\mathsf{K}-1)-\sum_{q\in[i-1]} n_{q,k}} = \cdots = d^k_{\mathsf{K}+i(\mathsf{K}-1)-\sum_{q\in[i]} n_{q,k}} = i. \tag{23}$$

For example, when $i = 1$, we let

$$d^k_{\mathsf{K}+1} = \cdots = d^k_{2\mathsf{K}-n_{1,k}-1} = 1,$$

[3]The order of the multicast messages in P_1 is not important because this order is generated randomly. Here, we assume this order for sake of easy comparison with the demand vector $(1, 2)$.

when $i = 2$, we let

$$d^k_{2\mathsf{K}-n_{1,k}} = \cdots = d^k_{3\mathsf{K}-n_{1,k}-n_{2,k}-2} = 2,$$

and so on. Hence, each file is requested by $\mathsf{K}-1$ effective users in the user set $[(\mathsf{K}-1)(\mathsf{N}-1)+\mathsf{K}]\backslash\{k\}$. For each file, we randomly and uniformly choose an effective user demanding this file as a leader user. The leader set is denoted by $\mathcal{L}_k$.

We generate a random permutation of $[(\mathsf{K}-1)(\mathsf{N}-1)+\mathsf{K}]\setminus\{k\}$, denoted by $\mathbf{q}_k = (q_{k,1}, \ldots, q_{k,\mathsf{U}})$, independently and uniformly over the set of all possible permutations.

For each set $\mathcal{S} \subseteq [\mathsf{U}]$ where $|\mathcal{S}| = t$, by computing $\mathcal{S}' = \cup_{j'\in\mathcal{S}}\{q_{k,j'}\}$, we generate the multicast message $W^k_{\mathcal{S}} = \underset{j\in\mathcal{S}}{\oplus} f^k_{d^k_{q_{k,j}},\mathcal{S}'\backslash\{q_{k,j}\}}$ as in (15). The trusted server asks user k to broadcast $X_k = (\mathscr{M}(P_k), P_k)$ to other users, where

$$P_k = \left(W^k_{\mathcal{S}} : (\cup_{j'\in\mathcal{S}}\{q_{k,j'}\}) \cap \mathcal{L} \neq \emptyset\right), \tag{24}$$

Notice that in the metadata of $W^k_{\mathcal{S}}$, the set $\mathcal{S}$ is revealed.

Decodability. We focus on user $k \in [\mathsf{K}]$. In the j^{th} transmission where $j \in [\mathsf{K}]\setminus\{k\}$, it was shown in [4, Lemma 1], user k can reconstruct each multicast message $W^j_{\mathcal{S}}$ where $\mathcal{S} \subseteq [\mathsf{U}]$ and $|\mathcal{S}| = t$. User k then checks each $W^j_{\mathcal{S}}$ where $\mathcal{S} \subseteq [\mathsf{U}]$ and $|\mathcal{S}| = t$. If $W^j_{\mathcal{S}}$ contains $t-1$ cached subfiles and one uncached subfile, user k knows this message is useful to it and decodes the uncached subfile.

It is obvious that each subfile of F_{d_k} which is not cached by user k, appears in one multicast message. Hence, after considering all transmitted packets in the delivery phase, user $k \in [\mathsf{K}]$ can recover all requested subfiles to reconstruct its requested file.

Privacy. The intuition on the privacy is the same as the above example and the information-theoretic proof on the privacy can be found in [12].

Performance. Each user $k \in [\mathsf{K}]$ broadcasts $\binom{\mathsf{U}}{t} - \binom{\mathsf{U}-\mathsf{N}}{t}$ multicast messages, each of which contains $\frac{\mathsf{B}}{\mathsf{K}\binom{\mathsf{U}}{t-1}}$ bits. Hence, the achieved load coincides with (11).

IV. Numerical Evaluations

We provide numerical evaluations of the proposed private caching schemes for the $(\mathsf{K},\mathsf{N},\mathsf{M}) = (10,5,\mathsf{M})$ D2D caching system with private demands. We compare the baseline D2D uncoded private caching scheme in Theorem 1 and the coded caching scheme in Theorem 2, with the converse bound in [3] for the shared-link caching model. It shows that the proposed coded caching scheme outperforms the uncoded scheme.

V. Conclusions

We introduced a novel D2D private caching model with a trusted server, which aims to preserve the privacy of the users demands. We proposed a novel D2D private coded caching scheme, which is order optimal within a factor of 6 when $\mathsf{N} \geq \mathsf{K}$ and $\mathsf{M} \geq 2\mathsf{N}/\mathsf{K}$, and within a factor of 12 when $\mathsf{N} < \mathsf{K}$ and $\mathsf{M} \geq \mathsf{N}/\mathsf{K}$. This scheme is also order optimal within a factor of 18 for any system parameters under the constraint of uncoded cache placement and privacy against colluding users.

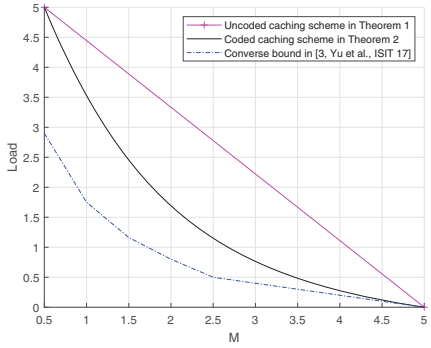

Fig. 2: (M,R) tradeoff for the $(\mathsf{K},\mathsf{N},\mathsf{M}) = (10,5,\mathsf{M})$ D2D caching system with private demands.

Acknowledgemnt

The work of K. Wan and G. Caire is supported by the European Research Council under the ERC Advanced Grant N. 789190, CARENET. The work of D. Tuninetti is supported in part by the National Science Foundation Award 1910309. The work of M. Ji is supported in part by NSF Awards 1817154 and 1824558.

References

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

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

[3] 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.

[4] ——, "The exact rate-memory tradeoff for caching with uncoded prefetching," *IEEE Trans. Infor. Theory*, vol. 64, pp. 1281 – 1296, Feb. 2018.

[5] 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.

[6] C. Yapar, K. Wan, R. F. Schaefer, and G. Caire, "On the optimality of d2d coded caching with uncoded cache placement and one-shot delivery," *in IEEE Int. Symp. Inf. Theory*, Jul. 2019.

[7] 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.

[8] K. Wan and G. Caire, "On coded caching with private demands," *arXiv:1908.10821*, Aug. 2019.

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

[10] V. R. Aravind, P. Sarvepalli, and A. Thangaraj, "Subpacketization in coded caching with demand privacy," *arXiv:1909.10471*, Sep. 2019.

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

[12] 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*, Dec. 2019.