Novel Converse for Device-to-Device Demand-Private Caching with a 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*—This paper considers cache-aided device-to-device (D2D) networks where a *trusted server* helps to preserve the privacy of the users' demands. Specifically, the trusted server collects the users' demands before the delivery phase and sends a query to each user, who then broadcasts multicast packets according to this query. Recently the Authors proposed a D2D private caching scheme that was shown to be order optimal except for the very low memory size regime, where the optimality was proved by comparing to a converse bound without privacy constraint. The main contribution of this paper is a novel converse bound for the studied model where users may collude (i.e., some users share cache contents and demanded files, and yet cannot infer what files the remaining users have demanded) and under the placement phase is uncoded. To the best of the Author's knowledge, such a general bound is the first that genuinely accounts for the demand privacy constraint. The novel converse bound not only allows to show that the known achievable scheme is order optimal in all cache size regimes (while the existing converse bounds cannot show it), but also has the potential to be used in other variants of demand private caching.**

I. Introduction

Coded caching was originally proposed by Maddah-Ali and Niesen (MAN) for shared-link networks [1]. In the MAN model, a server has access to a library of N equal-length files and is connected to K users through an error-free broadcast link. Each user can store up to M files in its cache. The MAN caching scheme includes a *placement* phase and a *delivery* phase that are designed so as to minimize the *worst-case load* (i.e., the number of files sent on the shared link that suffices to satisfy every possible demand vector). For the successful decoding of a MAN multicast message, the users need to know the composition of this message (i.e., which bits are coded together). As a consequence, users are aware of the demands of other users. In practice, schemes that leak information on the demand of a users to other users are highly undesirable. Shared-link coded caching with private demands, which aims to preserve the privacy of the users' demands from other users, was originally discussed in [2] and recently analyzed information-theoretically by Wan and Caire in [3]. It was shown in [3] that the privacy of the users' demand can be preserved by introducing virtual users. Compared to converse bounds for the shared-link model without privacy constraint from [4], this shared-link private scheme is order optimal in all regimes, except for $\mathsf{K} < \mathsf{N}$ (i.e., less users than files) and $\mathsf{M} < \frac{\mathsf{N}}{\mathsf{K}}$ (i.e., small memory regime) [3]. To the best of our knowledge, the only converse bound that accounts for privacy constraints was proposed in [5] for the case $\mathsf{K} = \mathsf{N} = 2$ only. By combining the novel private converse with existing non-private ones, the exact optimality for $\mathsf{K} = 2, \mathsf{N} = 2$ was characterized in [5], which curiously coincides with the optimal non-private scheme for $\mathsf{K} = 3, \mathsf{N} = 2$ [6].

In practice, the content of the library may have been already distributed across the users' local memories and can thus be delivered locally through peer-to-peer or Device-to-Device (D2D) communications. The shared-link model was extended to D2D networks in [7]. To preserve the privacy of the users' demands, a novel cache-aided D2D architecture with a *trusted server* was formulated in [8]. This trusted server is connected to each user through an individual secure link and without access to the library. 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, who then broadcasts packets accordingly. The objective is to design a two-phase private D2D caching scheme with minimum number of transmitted bits by all users in the delivery phase, while preserving the privacy of the demand of each user from the other users. By extending the virtual user strategy in [3], we proposed a novel D2D private caching scheme in [8]. By comparing the private achievable scheme in [8] to an existing converse bound that does not account for the demand privacy constraint, the private achievable scheme in [8] was shown to be order optimal in all regimes, except for $\mathsf{K} < \mathsf{N}$ (i.e., less users than files) and $\mathsf{M} \in [\mathsf{N}/\mathsf{K}, 2\mathsf{N}/\mathsf{K})$ (i.e., small memory regime).[1]

On the observation that the small memory regime with less users than files is open, the contributions of this paper are:

1) $\mathsf{N} \geq \mathsf{K} = 2$: we propose a novel converse bound under the constraint of uncoded cache placement that fully considers the privacy constraint; the bound is inspired by converse bounds for non-private shared-link caching models under uncoded placement [9], [10] and for Private Information Retrieval (PIR) systems [11]. In addition, we propose a novel two-user D2D private caching scheme,

[1]For example, when $\mathsf{K} < \mathsf{N}$ and $\mathsf{M} = \mathsf{N}/\mathsf{K}$, the gap between the private achievable scheme and an existing non-private converse bound is $\mathsf{N}/(\mathsf{K}-1)$, which can be unbounded.

978-1-7281-6432-8/20/$31.00 ©2020 IEEE

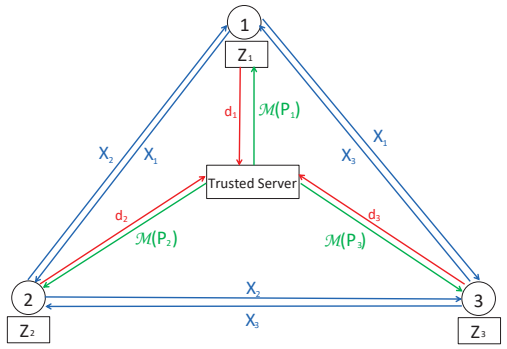

Fig. 1: The D2D private caching problem with a trusted server and $\mathsf{K} = 3$ users.

which strictly improves the existing scheme in [8]. Compared to this novel converse bound, the novel scheme is exactly optimal when the cache size is either small or large, and it is order optimal to within a factor of 3 otherwise (numerical simulations suggest $4/3$).

2) $\mathsf{N} \geq \mathsf{K} > 2$: we extend the novel converse bound to any number of users who may be colluding. By a cut-set idea, we divide the users into two groups and leverage the converse bound for the two-user case. Under the constraint of uncoded cache placement and privacy against colluding users, our D2D private caching scheme in [8] is shown to be order optimal to within a factor of 18 (numerical simulations suggest $27/2$) in all cache size regimes, also in the one left open in [8].

Paper Organization: Section II defines the problem. Section III presents the main results of this paper. Section IV provides some numerical evaluations and concludes the paper. Detailed proofs can be found in the online extended version of this paper [12].

Notation Convention: Calligraphic symbols denote sets, bold symbols denote vectors, and sans-serif symbols denote system parameters. Lower-case symbols denote realizations of random variables indicated with upper-case symbols. We use $|\cdot|$ to represent the cardinality of a set or the length of a vector. Sets of consecutive integers are denoted as $[a:b] := \{a, a+1, \ldots, b\}$ and $[n] := [1, 2, \ldots, n]$. $\binom{x}{y} = 0$ if $x < 0$ or $y < 0$ or $x < y$.

II. 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 independent files, denoted by $(F_1, F_2, \ldots, F_{\mathsf{N}})$, where each file is composed of B i.i.d. bits. 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. 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 files, and $\mathscr{M}(C_k)$ represents the metadata/composition of C_k (i.e., how C_k is generated). We have

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

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 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

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

Objective. We say that load R is achievable if

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

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 $\mathsf{R}^\star$.

Uncoded Cache Placement. If each user directly copies some bits of the files directly into its cache, the cache placement is said to be *uncoded*. The minimum load under the constraint of uncoded cache placement is denoted by $\mathsf{R}_{\mathrm{u}}^\star$.

Colluding Users. 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

$$I\big(\mathbf{D};\mathbf{X},\{Z_k:k\in\mathcal{S}\}|\{D_k:k\in\mathcal{S}\}\big)=0,\ \forall\mathcal{S}\subseteq[\mathsf{K}],\mathcal{S}\neq\emptyset. \tag{8}$$

The optimal load under the constraint of uncoded cache placement and the privacy constraint in (8) is denoted by $\mathsf{R}^\star_{\mathrm{u,c}}$.

Obviously, $\mathsf{R}^\star_{\mathrm{u,c}}\geq\mathsf{R}^\star_{\mathrm{u}}\geq\mathsf{R}^\star$. For $\mathsf{K}=2$, the constraints in (6) and (8) are equivalent, and thus we have $\mathsf{R}^\star_{\mathrm{u,c}}=\mathsf{R}^\star_{\mathrm{u}}$.

III. Main Results

A. Past work for any K

In [8], based on the idea of 'pretending' there are virtual-user in the system so as to confuse the users about their demand, we showed that the lower convex envelope of the following memory-load tradeoff points is achievable

$$(\mathsf{M},\mathsf{R})=\left(\frac{\mathsf{N}+t-1}{\mathsf{K}},\frac{\binom{\mathsf{U}}{t}-\binom{\mathsf{U}-\mathsf{N}}{t}}{\binom{\mathsf{U}}{t-1}}\right),\ \forall t\in[\mathsf{U}+1], \tag{9}$$

where $\mathsf{U}:=(\mathsf{K}-1)\mathsf{N}$ is the total number (real+virtual) users. In [8, Theorem 3] we showed that the scheme achieving (9) is order optimal to within a factor of 6 if $\mathsf{N}>\mathsf{K}$ and $\mathsf{M}\geq 2\mathsf{N}/\mathsf{K}$ (i.e., the small cache size regime is open), and to within a factor of 12 if $\mathsf{N}\leq\mathsf{K}$. One can check that this scheme satisfies the robust privacy constraint in (8) against colluding users.

This order optimality result was derived by comparing (9) with existing converse bounds without privacy constraint. The problem in the regime the regime $\mathsf{N}>\mathsf{K}$ and $\mathsf{M}\in[\mathsf{N}/\mathsf{K},2\mathsf{N}/\mathsf{K})$ can be intuitively understood as follows: for $\mathsf{M}=\mathsf{N}/\mathsf{K}$ the achievable load in (9) is N while the converse bound without privacy constraint is $\mathsf{K}-1$; the ratio of this two numbers can be unbounded. Hence, we are motivated to derive a novel converse bound by fully incorporating the privacy constraint in the small memory regime with less users than files.

B. Novel achievable scheme for $\mathsf{K}=2$

When $\mathsf{K}=2$, we observe that in the D2D private scheme in [8] some cached contents are redundant; by removing those redundancies we derive the following new scheme.

Theorem 1 (Novel scheme for two-user systems)**.** *For the $(\mathsf{K},\mathsf{N},\mathsf{M})=(2,\mathsf{N},\mathsf{M})$ D2D private caching system, $\mathsf{R}^\star_{\mathrm{u}}=\mathsf{R}^\star_{\mathrm{u,c}}$ is upper bounded by the lower convex envelope of the following memory-load tradeoff points*

$$\left(\frac{\mathsf{N}}{2}+\frac{\mathsf{N}t'}{2(\mathsf{N}+t'-1)},\frac{\mathsf{N}(\mathsf{N}-1)}{(t'+1)(\mathsf{N}+t'-1)}\right), \tag{10}$$

where $t'\in[0:\mathsf{N}-1]$ and the point $(\mathsf{N},0)$. □

The detailed description of the scheme can be found in the extended version of this paper in [12, Section IV-B]. Moreover, in [12, Appendix F] we show that Theorem 1 is strictly better than the scheme in (9) for two-user systems.

C. Example of our novel converse for $(\mathsf{K},\mathsf{N},\mathsf{M})=(2,2,6/5)$

Our converse bound is the key novelty of this paper. It truly accounts for the privacy constraint. The key is to derive several bounds that contain a 'tricky' entropy term that needs to be bounded in a non-trivial way; in some bounds this entropy term appears with a positive sign and in others with a negative sign; by linearly combining the bounds, the 'tricky' entropy term cancels out. Different from [5] for the shared-link caching with private demands for $\mathsf{N}=\mathsf{K}=2$, our converse bound focuses on the uncoded cache placement and works for any system parameters where $\mathsf{N}\geq\mathsf{K}=2$.

We start with an example to illustrate in the simplest possible setting the novel ideas needed to derive our converse bound that incorporates privacy, the $(\mathsf{K},\mathsf{N},\mathsf{M})=(2,2,6/5)$ D2D private caching system. In this case, Theorem 1 achieves load $7/5$. The converse bound under the constraint of uncoded cache placement for D2D caching without privacy in [13] gives load of $4/5$. In the following, we prove that the load $7/5$ is actually optimal for our D2D private caching problem under the constraint of uncoded cache placement.

Assume we have a working system, that is, a system where all encoding, decoding and privacy constraints listed in Section II are met. With a slight abuse of notation, a set operation over cache configurations is meant to represent the set operation over the cached information bits only, i.e., excluding metadatas. In addition, each notation of a set or a vector of bits also includes the metadata for these bits. In the following, in order not to clutter the derivation with unnecessary 'epsilons and deltas', we shall neglect the terms (such as metadatas, etc) that contribute $\epsilon_{\mathsf{B}}=o(\mathsf{B})$ when $\mathsf{B}\to\infty$ to a bound like the one in (13).

Without loss of generality (see [12, Remark 5]), each user caches a fraction $\mathsf{M}/\mathsf{N}=3/5$ of each file and each bit in the library is cached by at least one user. Assume that the cache configurations of the two users are Z_1^1 and Z_2^1, where $Z_1^1\cup Z_2^1=\{F_1,F_2\}$ For the demand vector $(d_1,d_2)=(1,1)$, any working scheme must produce transmitted signals (X_1,X_2) such that the demand vector $(d_1,d_2)=(1,1)$ can be satisfied. The following observation is critical: because of the privacy constraint, *from the viewpoint of user* 1*, there must exist a cache configuration of user* 2*, denoted by Z_2^2, such that $Z_1^1\cup Z_2^2=\{F_1,F_2\}$, $H(X_2|Z_2^2,\mathscr{M}(P_2))=0$, and F_2 can be decoded from (X_1,Z_2^2)*. If such a cache configuration Z_2^2 did not exist, then user 1 would know that the demand of user 2 is F_1 from (Z_1^1,X_1,X_2,d_1), which is impossible in a working private system. Similarly, from the viewpoint of user 2, there must exist a cache configuration of user 1, denoted by Z_1^2, such that $Z_1^2\cup Z_2^1=\{F_1,F_2\}$, $H(X_1|Z_1^2,\mathscr{M}(P_1))=0$, and F_2 can be decoded from (X_2,Z_1^2).

From (Z_1^1,Z_2^1), for each file $F_i,\ i\in\{1,2\}$, we have

$$|F_i\cap Z_1^1|=\frac{\mathsf{BM}}{\mathsf{N}}=\frac{3\mathsf{B}}{5}, \tag{11a}$$

$$|F_i\setminus Z_1^1|=|F_i\setminus Z_2^1|=\mathsf{B}-\frac{3\mathsf{B}}{5}=\frac{2\mathsf{B}}{5}, \tag{11b}$$

$$|F_i\cap Z_1^1\cap Z_2^1|=\frac{\mathsf{B}}{5}. \tag{11c}$$

Similarly, since $Z_1^1 \cup Z_2^1 = Z_1^1 \cup Z_2^2 = \{F_1, F_2\}$, we also have

$$|F_i \cap Z_1^1 \cap Z_2^2| = \frac{\mathsf{B}}{5}, \tag{11d}$$

$$F_i \setminus Z_1^1 \subseteq F_i \cap Z_2^1 \cap Z_2^2. \tag{11e}$$

Inspired by the genie-aided converse bound for shared-link caching networks without privacy in [9], [10], we construct a genie-aided super-user with cache content

$$Z' = \big(Z_2^1,\ Z_2^2 \setminus (F_1 \cup Z_2^1)\big), \tag{12}$$

who is able to recover the whole library from (X_1, Z'). Indeed, after file F_1 is reconstructed from (X_1, Z_2^1), the combination of $(F_1 \cup Z_2^1)$ and $Z_2^2 \setminus (F_1 \cup Z_2^1)$ gives Z_2^2; now, file F_2 can be reconstructed from (X_1, Z_2^2). Therefore, we have

$$\begin{aligned}
2\mathsf{B} &= H(F_1, F_2) \le H\big(X_1, Z'\big) && (13a)\\
&= H\big(X_1, Z_2^1, Z_2^2 \setminus (F_1 \cup Z_2^1)\big) && (13b)\\
&= H\big(X_1, Z_2^1\big) + H\big(Z_2^2 \setminus (F_1 \cup Z_2^1) | X_1, Z_2^1, F_1\big) && (13c)\\
&\le H(X_1) + H(Z_2^1) + H\big(Z_2^2 | Z_2^1, F_1\big) && (13d)\\
&= H(X_1) + H(Z_2^1) + H(F_2 \cap Z_2^2 \cap Z_1^1 | Z_2^1). && (13e)\\
&= H(X_1) + \underbrace{H(Z_2^1)}_{\le \mathsf{MB}} + \underbrace{H(F_2 \cap Z_2^2 \cap Z_1^1)}_{\le \mathsf{B}/5}\\
&\quad - H(\underbrace{F_2 \cap Z_2^2 \cap Z_1^1 \cap Z_2^1}_{:=\mathcal{Q}}), && (13f)
\end{aligned}$$

where (13e) follows because, in the last term of (13d), given F_i only the bits in F_2 are left, and because $Z_2^2 \setminus Z_2^1 = (Z_2^2 \cap Z_1^1) \setminus Z_2^1$ following the reasoning leading to (11e); the last step in (13f) follows because the bits in a file are independent.

At this point, we need a bound that can be combined with the one in (13) such that it contains on the right hand side the term $H(X_2)$, so that $H(X_1) + H(X_2)$ can be bounded by BR_{u}, and a term that allows one to get rid of the negative entropy of the random variable

$$\mathcal{Q} := F_2 \cap Z_1^1 \cap Z_2^1 \cap Z_2^2. \tag{14}$$

Next, we construct another genie-aided super-user in order to derive an inequality eliminating $H(\mathcal{Q})$ in (13f). We then focus on cache configurations Z_1^1 and Z_1^2, and the transmitted signal X_2. Recall that F_1 can be reconstructed from (Z_1^1, X_2), and F_2 can be reconstructed from (Z_1^2, X_2). Furthermore, by recalling the definition of $\mathcal{Q}$ in (14), it can be seen that the bits in $(F_2 \cap Z_1^1) \setminus \mathcal{Q}$ are independent of X_2. Hence, F_1 can be reconstructed from $(Z_1^1 \cap F_1, \mathcal{Q}, X_2)$. Hence, we can construct a super-user with cache content

$$Z'' = (Z_1^1 \cap F_1, Z_1^2 \cap F_2, \mathcal{Q}), \tag{15}$$

who can decode both files. Thus

$$\begin{aligned}
2\mathsf{B} &= H(F_1, F_2) \le H(X_2, Z'') && (16a)\\
&\le H(X_2) + \underbrace{H(Z_1^1 \cap F_1)}_{\le 3\mathsf{B}/5} + \underbrace{H(Z_1^2 \cap F_2)}_{\le 3\mathsf{B}/5} + H(\mathcal{Q}). && (16b)
\end{aligned}$$

Finally, by summing (13f) and (16b), we have that any achievable rate under uncoded cache placement must satisfy

$$\mathsf{R}_{\mathrm{u}} \ge \frac{H(X_1) + H(X_2)}{\mathsf{B}} \ge \frac{7}{5}. \tag{17}$$

The bound in (17) shows that Theorem 1 is indeed optimal for the considered memory point under the constraint of uncoded cache placement.

D. Results for $\mathsf{K} = 2$: novel converse bound and optimality

The key take-away points from the example in Section III-C are as follows:

1) By exploiting the privacy constraints, we note that from the viewpoint of each user k (i.e., given cache Z_k and transmitted packets (X_1, X_2)), any demand of the other user is equally possible. Hence, there must exist a cache configuration of the other user that allow for the decoding of any file using the same (X_1, X_2).
2) We introduce an auxiliary random variable $\mathcal{Q}$ to represents the set of bits $F_2 \cap Z_1^1 \cap Z_2^1 \cap Z_2^2$. We then use two different approaches to construct genie-aided super-users to decode the whole library, in such a way that we can get rid of tricky entropy term $H(\mathcal{Q})$ when the various bounds are summed together. In particular:
   a) In the first approach, we focus on (X_1, Z_2^1, Z_2^2) and construct a genie-aided super-user who can reconstruct the whole library by receiving X_1. The bits in $\mathcal{Q}$ belong to the overlap of Z_2^1 and Z_2^2. Hence, the size of the genie-aided super-user's cache decreases when $|\mathcal{Q}|$ increases. In other words, the load increases when $|\mathcal{Q}|$ increases (see (13f)).
   b) In the second approach, we focus on (X_2, Z_1^1, Z_1^2) and construct a genie-aided super-user who can reconstruct the whole library by receiving X_2. Now the bits in $\mathcal{Q}$ are in the cache of the super-user. Hence, the size of the genie-aided super-user's cache increases when $|\mathcal{Q}|$ increases. In other words, the load decreases when $|\mathcal{Q}|$ increases (see (16b)).

   Finally, by summing (13f) and (16b), the effect of $\mathcal{Q}$ is fully cancelled, such that we derive (17).

In [12, Section V-A], we generalize the example in Section III-C to any $\mathsf{N} \ge \mathsf{K} = 2$ and show:

Theorem 2 (Novel converse bound for two-user systems)**.** *For the $(\mathsf{K}, \mathsf{N}, \mathsf{M})$ D2D private caching system where $\mathsf{N} \ge \mathsf{K} = 2$, assuming $\mathsf{M} = \frac{\mathsf{N}}{2} + y$ where $y \in \big[0, \frac{\mathsf{N}}{2}\big]$, we have*

$$\begin{aligned}
\mathsf{R}_{\mathrm{u}}^\star &\ge \mathsf{N} - 2y - \frac{4y + (\mathsf{N} - \mathsf{K}/2)h}{h+2}\\
&\quad + \frac{2y}{\mathsf{N}} \frac{h^2(\mathsf{N} - \mathsf{K}/2) - \mathsf{N}(2\mathsf{N}/\mathsf{K} - 3) + h(\mathsf{N} + \mathsf{K}/2)}{(h+1)(h+2)}, && (18)\\
\mathsf{R}_{\mathrm{u}}^\star &\ge \mathsf{K}\left(1 - \frac{3y}{\mathsf{N}}\right), && (19)\\
\mathsf{R}_{\mathrm{u}}^\star &\ge \mathsf{K}\left(\frac{1}{2} - \frac{y}{\mathsf{N}}\right), && (20)
\end{aligned}$$

for $h \in [0 : \mathsf{N} - 3]$. □

By comparing the novel converse bound in Theorem 2 and the achievable scheme in Theorem 1, we have the following performance guarantees under the constraint of uncoded cache placement (the proofs can be found in [12, Appendix G]).

Theorem 3 (Optimality for two-user systems)**.** *For the* $(\mathsf{K},\mathsf{N},\mathsf{M})$ *D2D private caching system where* $\mathsf{N}\geq\mathsf{K}=2$*, the scheme in Theorem 1 is optimal under the constraint of uncoded cache placement when* $\frac{\mathsf{N}}{2}\leq\mathsf{M}\leq\frac{\mathsf{N}+1}{2}$ *or* $\frac{\mathsf{N}(3\mathsf{N}-5)}{2(2\mathsf{N}-3)}\leq\mathsf{M}\leq\mathsf{N}$.

In general, under the constraint of uncoded cache placement, the scheme in Theorem 1 is order optimal to within a factor of 3 *(numerical simulations suggest* $4/3$*).* □

From Theorem 3, we directly derive the following corollary.

Corollary 1. *For the* $(\mathsf{K},\mathsf{N},\mathsf{M})$ *D2D private caching system where* $\mathsf{K}=2$ *and* $\mathsf{N}\in\{2,3\}$*, the scheme in Theorem 1 is optimal under the constraint of uncoded cache placement.* □

E. Results for any K*: novel converse bound and optimality when users may collude*

In [12, Section V-B], we extend Theorem 2 to any $\mathsf{K}\geq 2$ with the consideration of the privacy constraint against colluding users in (8). The main idea is to divide the users into two groups, and for each group generate a powerful aggregated user whose cache contains the caches of all users in each group (implying collusion). When $\mathsf{K}/2$ and $2\mathsf{N}/\mathsf{K}$ are integers, the above genie-aided system is equivalent to the two-user D2D private caching problem with $2\mathsf{N}/\mathsf{K}$ files, each of which has $\mathsf{K}\mathsf{B}/2$ bits, and each of the two users caches $\left(\frac{\mathsf{N}\mathsf{B}}{2}+y\mathsf{B}\right)$ bits in its cache and demands one file. When $\mathsf{K}/2$ or $2\mathsf{N}/\mathsf{K}$ are not integers, some additional steps are needed as described in [12, Appendix C]. The resulting converse bound is as follows.

Theorem 4 (Novel converse bound for K-user systems)**.** *For the* $(\mathsf{K},\mathsf{N},\mathsf{M})$ *D2D private caching system where* $\mathsf{N}\geq\mathsf{K}\geq 3$*, assuming* $\mathsf{M}=\frac{\mathsf{N}}{\mathsf{K}}+\frac{2y}{\mathsf{K}}$ *where* $y\in\left[0,\frac{\mathsf{N}}{2}\right]$*, we have*

$$\mathsf{R}^{\star}_{\mathrm{u,c}}\geq\frac{\lfloor\mathsf{K}/2\rfloor}{\lceil\mathsf{K}/2\rceil}\frac{\lfloor 2\mathsf{N}/\mathsf{K}\rfloor}{2\mathsf{N}/\mathsf{K}}\times\text{'RHS eq(18)'},\quad h\in[0:\lfloor 2\mathsf{N}/\mathsf{K}-3\rfloor],\tag{21}$$

$$\mathsf{R}^{\star}_{\mathrm{u,c}}\geq\frac{\lfloor\mathsf{K}/2\rfloor}{\lceil\mathsf{K}/2\rceil}\times\text{'RHS eq(19)'},\tag{22}$$

$$\mathsf{R}^{\star}_{\mathrm{u,c}}\geq\frac{\lfloor\mathsf{K}/2\rfloor}{\lceil\mathsf{K}/2\rceil}\times\text{'RHS eq(20)'},\tag{23}$$

where 'RHS eq(n*)' stands for 'right hand side of equation number* n*'.* □

By comparing our D2D private caching scheme in (9) and the combination of the novel converse bound in Theorem 4 and the converse bound for shared-link caching without privacy in [9], we can characterize the order optimality under the constraint of uncoded cache placement as follows. The proof can be found in [12, Appendix H].

Theorem 5 (Order optimality for K-user systems)**.** *For the* $(\mathsf{K},\mathsf{N},\mathsf{M})$ *D2D private caching system where* $\mathsf{N}\geq\mathsf{K}$*, the scheme in* (9) *is order optimal under the constraint of uncoded cache placement and privacy against colluding users, within a factor of* 18 *(numerical simulations suggest* $27/2$*).* □

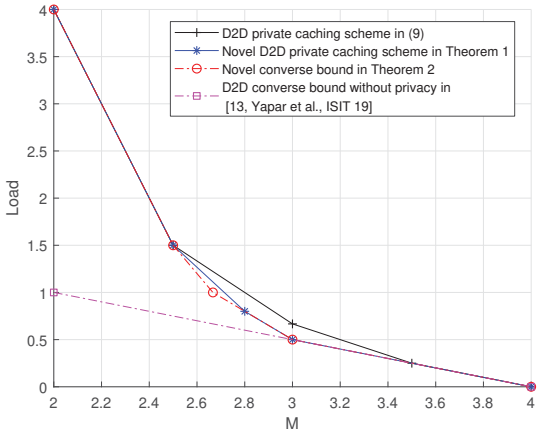

Fig. 2: The memory-load tradeoff for the D2D caching problem with private demands, where $\mathsf{K}=2$ and $\mathsf{N}=4$.

Notice that when $\mathsf{N}<\mathsf{K}$, it was proved in [8, Theorem 3] that the scheme in (9) is generally order optimal within a factor of 12. Hence, from Theorem 5, we can directly have the following conclusion.

Corollary 2. *For the* $(\mathsf{K},\mathsf{N},\mathsf{M})$ *D2D private caching system, the scheme in* (9) *is order optimal under the constraint of uncoded cache placement and privacy against colluding users, within a factor of* 18*.* □

IV. Numerical Evaluations and Conclusions

Numerical Evaluations: In Fig. 2, we consider the case where $\mathsf{K}=2$ and $\mathsf{N}=4$. For the achievable schemes we plot the D2D private caching scheme in [8] (reviewed in (9)) and the novel D2D private caching scheme for two-user systems in Theorem 1. For the converse bounds we plot the novel converse bound in Theorem 2 and the converse bound under the constraint of uncoded cache placement in [13] for D2D caching without privacy. Fig. 2 shows that the proposed caching scheme and the proposed converse bound meet for all memories except $\frac{5}{2}\leq\mathsf{M}\leq\frac{14}{5}$.

Conclusions: We considered D2D private caching with a trusted server, which aims to preserve the privacy of the users' demands in peer-to-peer cache-aided systems. We derived a novel converse bound under the constraint of uncoded cache placement and privacy against colluding users. Compared with the novel converse bound, an existing D2D private scheme was proved to be order optimal within a constant factor in any parameter regimes, even the small memory regime with less users than files that was open before this paper.

Acknowledgement: The work of K. Wan and G. Caire was partially funded by the European Research Council under the ERC Advanced Grant N. 789190, CARENET. The work of M. Ji was supported in part by NSF Awards 1817154 and 1824558. The work of D. Tuninetti was supported in part by NSF Award 1910309.

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] 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.

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

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

[5] S. Kamath, J. Ravi, and B. K. Dey, "Demand-private coded caching and the exact trade-off for n=k=2," *arXiv:1911.06995*, Nov. 2019.

[6] C. Tian, "Symmetry, demand types and outer bounds in caching systems," *in IEEE Int. Symp. Inf. Theory*, pp. 825–829, Jul. 2016.

[7] 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.

[8] K. Wan, H. Sun, M. Ji, D. Tuninetti, and G. Caire, "Device-to-device private caching with trusted server," *arXiv:1909.12748, submitted to ICC 20*, Sep. 2019.

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

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

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

[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.

[13] 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.