Fundamental Limits of Caching With Secure Delivery

Avik Sengupta, *Student Member, IEEE*, Ravi Tandon, *Member, IEEE*,
and T. Charles Clancy, *Senior Member, IEEE*

***Abstract*— Caching is emerging as a vital tool for alleviating the severe capacity crunch in modern content-centric wireless networks. The main idea behind caching is to store parts of the popular content in end-users' memory and leverage the locally stored content to reduce peak data rates. By jointly designing content placement and delivery mechanisms, recent works have shown order-wise reduction in transmission rates in contrast to traditional methods. In this paper, we consider the secure caching problem with the additional goal of minimizing information leakage to an external wiretapper. The fundamental cache memory versus transmission rate tradeoff for the secure caching problem is characterized. Rather surprisingly, these results show that security can be introduced at a negligible cost, particularly for large number of files and users. It is also shown that the rate achieved by the proposed caching scheme with secure delivery is within a constant multiplicative factor from the information-theoretic optimal rate for almost all parameter values of practical interest.**

***Index Terms*— Caching, information theoretic security, multicast delivery.**

I. INTRODUCTION

In MODERN content-centric wireless networks, caching helps in reducing the peak network load at times of high traffic volume. Fractions of popular content are stored locally in end-users' cache memories distributed across a given network. At times of high demand, the users can be partly served locally from their cache, thereby reducing the network load. Caching generally works in two phases - the *storage phase* and the *delivery phase*. The general caching problem has been well studied in literature [3]–[6]. Traditionally, the delivery phase of caching systems operate as a series of dedicated unicast transmissions to individual users by transmitting fractions of requested files which are not stored in their caches. However, this is not a scalable solution as the number of users in the system increases. A more efficient solution is to deliver content simultaneously to users through multicast transmissions. Most of the prior works in this area tend to use a fixed delivery scheme and then optimize the storage phase to suit the delivery scheme [5], [6]. Further, their investigations are mainly based on the gains obtained from local content distribution, ignoring the global cache interactions and content sharing as a factor for extracting caching gain.

More recently, [7]–[12] have proposed information theoretic formulations of the caching problem. In [7], a scheme is proposed which, in addition to the local caching gain, is also capable of offering a global caching gain. The scheme takes the cumulative size of the network cache memory into consideration and *jointly designs* the cache storage phase and a coded mutlicast delivery phase. This achieves a global caching gain which provides an order-wise improvement over local caching gain. The fundamental concepts presented in [7] are extended to the case of decentralized storage in [8] and non-uniform ZipF [13] user demands in [9] and [14]. Some extensions of the caching problem have been investigated in the case of Device-to-Device (D2D) communications in [15]–[18], from the perspective of content distribution networks in [19] and reinforcement learning in [20]–[22].

In this paper, we investigate the fundamental *security* aspects of the caching problem in the presence of an external adversary (wiretapper). To this end, we introduce the *secure caching problem* in which the multicast communication between the central server and the users (delivery phase) occurs over a *public (insecure) channel*. The defining feature of this problem is to capture the tradeoff between the multicast rate of the insecure link and the size of the cache memory. To the best of our knowledge, none of the works on cache storage and placement design deal with security issues. We consider a system with a central server connected to K users through an error-free rate-limited link. The server has a database of N files denoted by $(W_1, \ldots, W_N)$, where each file is of size F bits. For the scope of this paper, we assume that a user can request access to *any* one of the files at a given time. Each user has a cache memory Z_k of size MF bits for any real number $M \in [0, N]$. Similar to [7], the system operates over two phases: a cache *storage phase* and a *delivery phase*. The storage phase can be of two types: *centralized storage* or *decentralized storage*. In case of centralized storage, the central server stores the cache Z_k of user k with some content, which is a function of the files $(W_1, \ldots, W_N)$. In case of decentralized storage, the user k is allowed to store any random combination of bits from each file without coordination from the central server. User k (for $k = 1, \ldots, K$) then requests access to one of the files W_{d_k} in the database. In the delivery phase, the central server proceeds by transmitting a signal

Manuscript received January 31, 2014; revised July 11, 2014 and November 7, 2014; accepted November 12, 2014. Date of publication December 2, 2014; date of current version January 13, 2015. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Y.-W. P. Hong.

A. Sengupta and T. C. Clancy are with the Hume Center for National Security and Technology, Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061 USA (e-mail: aviksg@vt.edu; tcc@vt.edu).

R. Tandon is with the Discovery Analytics Center, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061 USA (e-mail: tandonr@vt.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2014.2375553

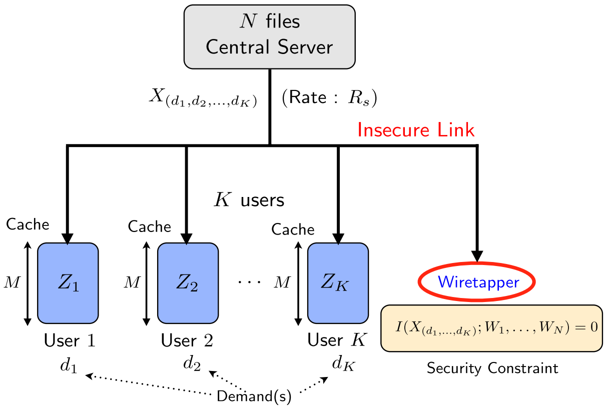


Fig. 1. System Model for Secure Caching.

$X_{(d_1,\dots,d_K)}$ of size RF bits over the shared link. Using the content Z_k (of its cache) and the received signal $X_{(d_1,\dots,d_K)}$, the $k-$th user intends to reconstruct the requested file W_{d_k}. A memory-rate pair (M, R) is *achievable* if for a (per-user) cache size of MF bits, and using rate RF bits, it is possible for each user to decode its requested file for *any* set of requests $(d_1, \dots, d_K)$. Let $R^*(M)$ denote the smallest rate R such that the pair (M, R) is achievable. The function $R^*(M)$ is the fundamental *memory-rate* tradeoff for the caching problem. An approximate characterization for $R^*(M)$ was provided in [7]–[9].

We consider this problem in the presence of an external wiretapper which can observe the multicast communication $X_{(d_1,\dots,d_K)}$ i.e., the communication from the central server to the users occurs over an *insecure* link. The wiretapper is considered to be strictly out-of-network and is thus able to observe only the multicast delivery which happens over a broadcast channel. Thus, besides satisfying the users' demands, we require that $X_{(d_1,\dots,d_K)}$ must not reveal any information about $(W_1, \dots, W_N)$ i.e., $I\left(X_{(d_1,\dots,d_K)}; W_1, \dots, W_N\right) = 0$. As is shown, the additional security constraint necessitates introducing randomness in the form of keys, which occupy a part of the cache of each user. Subsequently, these keys are used in the delivery phase to the keep the delivery information theoretically secure using a one-time-pad scheme [23]. In our system model, the placement phase occurs over unicast channels to individual users and can be secured with the help of individual keys e.g., secure unicast communications using a system similar to code-division-multiple-access (CDMA). As a result, security is considered to be inherent in the placement phase. Thus, in this work, we consider the security of only the *delivery phase* and not the cache *placement phase*. For this problem, a memory-rate pair (M, R_s) is *securely achievable* if, for a cache size of MF and a transmission of rate R_sF bits, it is possible for each user to decode its requested file and the communication over the shared link reveals no information about any file. Fig. 1 shows the caching system in the presence of a wiretapper. Let $R_s^*(M)$ denote the smallest R_s such that (M, R_s) is achievable. Thus, the function $R_s^*(M)$ is the fundamental memory-rate tradeoff for the *secure* caching problem. We investigate both the centralized cache placement as well as the decentralized placement with secure file delivery without any assumptions on user demands and file popularity.

The main contribution of this paper is an approximate characterization of $R_s^*(M)$. We design centralized and decentralized caching algorithms which make use of coded multicast delivery to extract global caching gain. The system has uniformly distributed orthogonal keys which are stored across users for secure multicast delivery. We present novel upper and lower bounds on $R_s^*(M)$ and show that these bounds are within a constant multiplicative gap. Indeed, for a fixed M, it is intuitively clear that $R_s^*(M) \geq R^*(M)$, i.e., the minimum rate in presence of a wiretapper must be, in general, larger than in the absence of a wiretapper. From our results, we show, rather surprisingly, that the cost for incorporating security in both the centralized and decentralized caching schemes is negligible when the number of users and files are large.

II. System Model

Let $(W_1, W_2, \dots, W_N)$ be N independent random variables each uniformly distributed over

$$[2^F] \triangleq \{1, 2, \dots, 2^F\} \tag{1}$$

for some $F \in \mathbb{N}$. Each W_n represents a file of size F bits. A (M, R_s) secure caching scheme comprises of K *random* caching functions, N^K *random* encoding functions and KN^K decoding functions. The K *random* caching functions map the files $(W_1, \dots, W_N)$ into the cache content:

$$Z_k \triangleq \phi_k\big(W_1, \dots, W_N\big) \tag{2}$$

for each user $k \in [K]$ during the storage (or placement) phase. The maximum allowable size of the contents of each cache Z_k is MF bits. The N^K *random* encoding functions map the files $(W_1, \dots, W_N)$ to the input

$$X_{(d_1,\dots,d_K)} \triangleq \psi_{(d_1,\dots,d_K)}\big(W_1, \dots, W_N\big) \tag{3}$$

of the shared link in response to the requests $(d_1, \dots, d_K) \in [N]^K$ during the delivery phase. Finally, the KN^K decoding functions map the received signal over the *insecure* shared link $X_{(d_1,\dots,d_K)}$ and the cache content Z_k to the estimate

$$\hat{W}_{(d_1,\dots,d_K),k} \triangleq \mu_{(d_1,\dots,d_K),k}\left(X_{(d_1,\dots,d_K)}, Z_k\right) \tag{4}$$

of the requested file W_{d_k} for user $k \in [K]$. The probability of error is defined as:

$$P_e \triangleq \max_{(d_1,\dots,d_K)\in[N]^K} \max_{k\in[K]} \mathbb{P}(\hat{W}_{(d_1,\dots,d_K),k} \neq W_{d_k}). \tag{5}$$

The information leaked at the wiretapper is defined as:

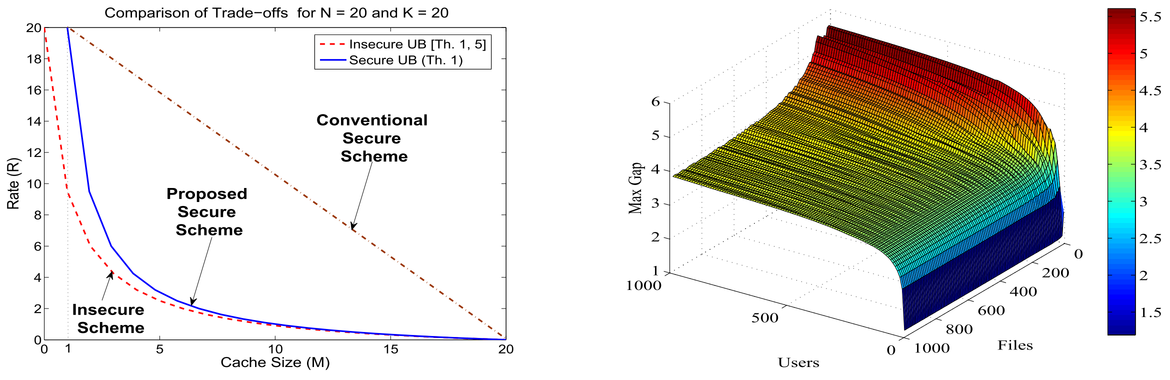
$$L \triangleq \max_{(d_1,\dots,d_K)\in[N]^K} I\left(X_{(d_1,\dots,d_K)}; W_1, \dots, W_N\right). \tag{6}$$

Definition 1: The pair (M, R_s) is *securely achievable* if for any $\epsilon > 0$ and every large enough file size F, there exists a (M, R_s) secure caching scheme with $P_e \leq \epsilon$ and $L \leq \epsilon$. We define the secure memory-rate tradeoff

$$R_s^*(M) \triangleq \inf\{R_s : (M, R_s) \textit{ is securely achievable}\}. \tag{7}$$

III. Centralized Caching With Secure Delivery

The first result gives an achievable rate which upper bounds the optimal memory-rate trade-off $R_s^*(M)$ for the centralized



(a) (b)

Fig. 2. (a) Centralized Secure vs. Non-Secure Bounds $N = K = 20$. (b) Multiplicative gap between $R_s^C(M)$ and lower bound on $R_s^*(M)$.

caching scheme with secure delivery. Security is incorporated by introducing randomness in the storage and delivery phase of the achievable scheme in form of a set of uniformly distributed orthogonal keys (independent of the data) stored in the cache of each user. The total cache memory (of size MF bits) is divided into two parts - data memory (of size $M_D F$ bits) and key memory (of size $M_K F$ bits) such that $M = M_D + M_K$. The server uses the keys stored at the users' caches to encode the delivery signal $X_{(d_1,\ldots,d_K)}$ such that the transmission is secure from the wiretapper.

Theorem 1: For N files and K users, each with a cache size of $M \in \frac{(N-1)}{K} \cdot t + 1$, for $t \in \{0, 1, 2, \ldots, K\}$ we have

$$R_s^*(M) \leq R_s^C(M) \triangleq K \cdot \left(1 - \frac{M-1}{N-1}\right) \left\{\frac{1}{1+K \cdot \frac{M-1}{N-1}}\right\} \tag{8}$$

i.e., the rate $R_s^C(M)$ is securely achievable. For any $1 \leq M \leq N$, the lower convex envelope of these points is achievable.

The algorithm achieving the rate in Theorem 1 is presented in Algorithm 1 (Appendix A). Similar to [7], the achievable rate in (8) consists of three factors. The first factor K is the worst case rate in the case when no data is cached ($M_D = 0$). The second factor in (8) is $\left(1 - \frac{M-1}{N-1}\right)$. This is the *secure local caching gain* and is relevant whenever M is of the order of N. The third factor in (8) is $1/\left(1 + K \cdot \frac{M-1}{N-1}\right)$, which is the *secure global caching gain*. Comparing Theorem 1 to ([7, Th. 1]), we observe that the terms $\frac{M}{N}$ in ([7, Th. 1]) have been replaced by $\frac{M-1}{N-1}$. However, the combination of the global and local gains leads to the rate in (8) being higher than the rate in ([7, Th. 1]) for a given value of M, N. This is the cost paid for the security in the system. However, as K, N become large, the secure rate is asymptotically equal to the non-secure case. When $N = K = 20$, it can be seen from Fig. 2(a) that the secure and non-secure bounds almost coincide i.e., security from a wiretapper can be achieved at *almost negligible cost* for a large number of files and users.

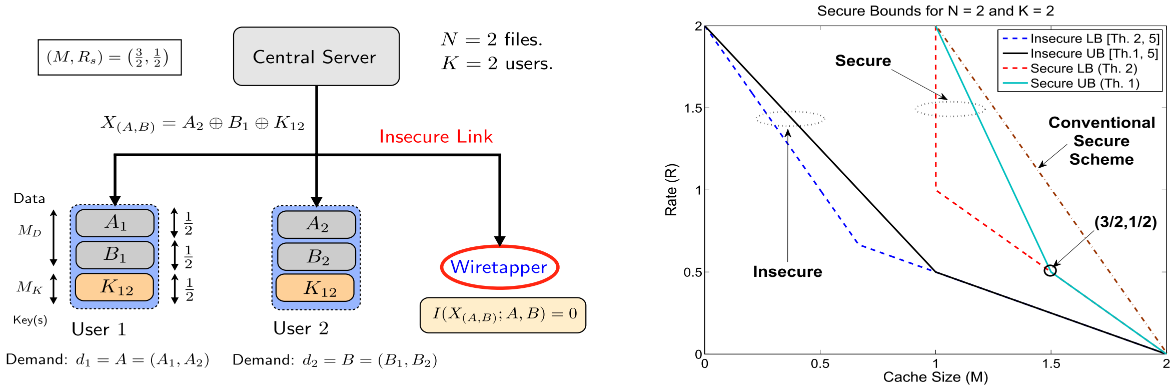
Algorithm 1 Secure Centralized Caching Algorithm

```
Centralized Cache Placement: for files $W_1, \ldots, W_N$
$t = K(M-1)/(N-1)$
for $n \in \{1, 2, \ldots, N\}$ do
    Split file $W_n$ into equal sized fragments $W_{n,\mathcal{T}} : \mathcal{T} \subseteq \{1, 2, \ldots, K\}, |\mathcal{T}| = t$
end for
Generate keys $\mathcal{K}_{\mathcal{T}_k}$ such that $\mathcal{T}_k \subseteq \{1, 2, \ldots, K\}, |\mathcal{T}_k| = t + 1$
for $k \in \{1, 2, \ldots, K\}$ do
    for $n = 1, 2, \ldots, N$ do
        File $W_{n,\mathcal{T}}$ is place in cache, $Z_k$, of user $k$ if $k \in \mathcal{T}$
        Key $\mathcal{K}_{\mathcal{T}_k}$ is placed in cache, $Z_k$, of user $k$ if $k \in \mathcal{T}_k$
    end for
end for
Coded Delivery:
for $\mathcal{S}$ such that $\mathcal{S} \subseteq \{1, 2, \ldots, K\}, |\mathcal{S}| = t + 1$ do
    Server sends $\left\{\mathcal{K}_{\mathcal{S}} \oplus_{k \in \mathcal{S}} W_{d_k, \mathcal{S} \setminus \{k\}}\right\}$
end for
```

Consider the case of conventional unicast content delivery to each user. In contrast to the insecure scheme in [7], to make the delivery phase secure, however, each user has to store a unique key (of the same size as a single file). During delivery, the server encodes the user's requested file with its key and transmits it. Thus, even with no data storage in cache, the cache size has to be at least F bits to store a key ($M_K = 1$) i.e., in the secure problem, $M = 0$ is infeasible. The worst case rate is achieved at $M = 1$ and the (M, R_s^C) pair $(1, K)$ is achievable. At the other extreme when $M = N$ i.e., the case where all files are stored in the user's cache and no content delivery is required. In this case $M_D = N, M_K = 0$ and the (M, R_s^C) pair $(N, 0)$ is achievable. We refer to a scheme which achieves points on the line joining $(1, K)$ and $(N, 0)$ as the *conventional secure scheme*, where each user stores one unique key and encrypted files are unicast to each user based on their request. On the other hand, the proposed scheme in Algorithm 1 jointly designs the placement of data and keys in the users' caches such that *coded secure multicasting* can be achieved among users. Next, we present a lower bound on $R_s^*(M)$ stated in the following theorem.

Theorem 2: For N files and K users, each having a cache size $1 \leq M \leq N$,

$$R_s^*(M) \geq \max_{s \in \{1,\ldots,\min\{N,K\}\}} \left(s - \frac{s(M-1)}{\left(\lfloor \frac{N}{s} \rfloor - 1\right)}\right). \tag{9}$$



(a)

(b)

Fig. 3. (a) Secure Caching Scheme and (b) (M, R_s^C) trade-off for $N = K = 2$.

The proof of Theorem 2 is presented in Appendix B. Next, we compare the achievable rate from Theorem 1 and the lower bound on the optimal rate in Theorem 2, and show that a constant multiplicative gap exists between $R_s^*(M)$ and the achievable rate $R_s^C(M)$.

Theorem 3: For N files and K users, each having a cache size $\max\left\{\frac{(K-N)(N-1)}{KN} + 1, 1\right\} \leq M \leq N$,

$$1 \leq \frac{R_s^C(M)}{R_s^*(M)} \leq 17. \tag{10}$$

The proof of Theorem 3 is presented in Appendix C. The gap is unbounded and scales with K only for the case of $K > N$ in the regime $1 \leq M < \frac{(K-N)(N-1)}{KN} + 1$, which is negligibly small for large K, N as discussed in Appendix C. While the analytical constant of 17 is large for practical purposes, the gap can tightened numerically. Fig. 2(b) shows the maximum value of the multiplicative gap between $R_s^C(M)$ and the lower bound on $R_s^*(M)$ for values for N, K ranging from 1 to 1000 and all feasible values of M in each case. It can be seen that the gap is generally less than 4 when $K < N$. However for $K > N$, and for small N, the gap is larger i.e., around 6.

A. Intuition Behind Theorem 1 (Achievability)

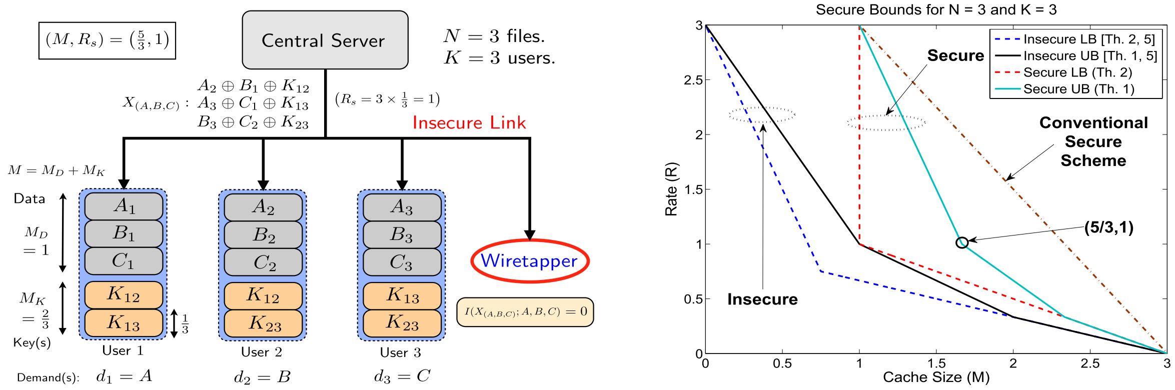
We next present a series of examples to explain the intuition behind the achievable rate in Theorem 1 and highlight the interesting features of the proposed secure delivery scheme.

Example 1: We illustrate the achievable scheme in Theorem 1 for the case of $N = 2$ files and $K = 2$ users. From Theorem 1 we have $M \in \frac{2-1}{2}\{0, 1, 2\} + 1 = \{1, \frac{3}{2}, 2\}$ are the possible cache sizes for each user. Let the two files be $W_1 = A$ and $W_2 = B$. The bounds in Theorems 1 and 2 are shown in Fig. 3(b) along with the bounds for the non-secure case from [7]. We start with the upper bound in Theorem 1. Considering the extreme point $M = 1$, the cache of both users Z_1, Z_2 only stores two unique keys $\mathcal{K}_1, \mathcal{K}_2$ and the server transmits both the files A, B over the shared link XOR-ed with a key. Given the worst-case demand $(d_1, d_2) = (A, B)$, the server can transmit $X_{(A,B)} = \{A \oplus \mathcal{K}_1, B \oplus \mathcal{K}_2\}$. This system satisfies every possible request with rate $R = 2$ and it is easily verified that $I\left(X_{(A,B)}; A, B\right) = 0$. Thus $(M, R_s^C) = (1, 2)$ is *securely* achievable. At the other extreme, when $M = 2$, each user can cache both files and no transmission is necessary. Hence the $(M, R_s^C) = (2, 0)$ is *securely* achievable.

Now we consider the intermediate case in which $M = 3/2$. The scheme for this scenario is depicted in Fig. 3(a). Both the files are split into 2 equal parts: $A = (A_1, A_2)$ and $B = (B_1, B_2)$, where A_1, A_2, B_1, B_2 are each of size $F/2$ bits. We also generate a key $\mathcal{K}_{12} \sim \text{unif}\{1, \ldots, 2^{(F/2)}\}$, which is independent of both the files A, B and has the same size as the sub-files i.e., $F/2$ bits. In the storage phase, the server fills the caches as follows: $Z_1 = (A_1, B_1, \mathcal{K}_{12})$ and $Z_2 = (A_2, B_2, \mathcal{K}_{12})$ i.e., each user stores one exclusive part of each file and the key. Thus $M_D = 1/2 + 1/2 = 1$ and $M_K = 1/2$. Now, consider the worst case request $(d_1, d_2) = (A, B)$. In order to satisfy this request, user 1 requires the file fragment A_2 while user 2 requires the file fragment B_1. In this case, the server transmits $X_{(A,B)} = \{A_2 \oplus B_1 \oplus \mathcal{K}_{12}\}$ which is of rate $1/2$. User 1 can obtain A_2 by XOR-ing out $B_1, \mathcal{K}_{12}$ while user 2 can get B_1 by XOR-ing out $A_2, \mathcal{K}_{12}$ from $X_{(A,B)}$. A wiretapper, on the other hand, would gain no knowledge of either file from the transmission since $I\left(X_{(A,B)}; A, B\right) = 0$ which follows from the fact that the key $\mathcal{K}_{12}$ is uniformly distributed. Thus, $(M, R_s^C) = (3/2, 1/2)$ is *securely* achievable. This can be seen in the secure upper bound in Fig. 3(b). Given that the points $(1, 2)$, $(3/2, 1/2)$ and $(2, 0)$ are achievable, the lines joining pairs of these points are also achievable. Thus, this proves the achievability of the secure upper bound in Fig 3(b). The gap between the insecure and secure achievable bounds results from the storage of the key in the users' cache. ◇

In the two user example, there is only a single key $\mathcal{K}_{12}$ in the system. Thus, if the key is compromised, the security of the entire system fails. The scheme proposed in Theorem 1 for general values of (N, K), however is more robust in its key management when the number of files and users increase. We next illustrate this point through an example.

Example 2: We consider the case for $N = K = 3$. For this case, from Theorem 1, $M \in \{1, \frac{5}{3}, \frac{7}{3}, 3\}$. The system and bounds for this case are illustrated in Fig. 4(a) and 4(b). We consider the case of $M = 5/3$ and three files A, B, C. Each file is split into 3 equal parts i.e., $A = (A_1, A_2, A_3)$, $B = (B_1, B_2, B_3)$, $C = (C_1, C_2, C_3)$. We also have 3 keys in the system, $\mathcal{K}_{12}, \mathcal{K}_{13}, \mathcal{K}_{23}$. In this case,


$(M, R_s) = (\frac{5}{3}, 1)$
Central Server
$N = 3$ files.
$K = 3$ users.
$A_2 \oplus B_1 \oplus K_{12}$
$X_{(A,B,C)}$: $A_3 \oplus C_1 \oplus K_{13}$
$B_3 \oplus C_2 \oplus K_{23}$
$(R_s = 3 \times \frac{1}{3} = 1)$
Insecure Link
$M = M_D + M_K$
Data
M_D $= 1$
M_K $= \frac{2}{3}$
Key(s)
$\frac{1}{3}$
A_1 B_1 C_1 K_{12} K_{13}
A_2 B_2 C_2 K_{12} K_{23}
A_3 B_3 C_3 K_{13} K_{23}
User 1
User 2
User 3
Demand(s): $d_1 = A$
$d_2 = B$
$d_3 = C$
Wiretapper
$I(X_{(A,B,C)}; A, B, C) = 0$
Secure Bounds for N = 3 and K = 3
Insecure LB [Th. 2, 5]
Insecure UB [Th. 1, 5]
Secure LB (Th. 2)
Secure UB (Th. 1)
Secure
Conventional Secure Scheme
(5/3,1)
Insecure
Rate (R)
Cache Size (M)
0
0.5
1
1.5
2
2.5
3

(a)

(b)

Fig. 4. (a) Secure Caching Scheme and (b) (M, R_s^C) trade-off for $N = K = 3$.

each subfile and each key is of size $F/3$ bits. In general, the key $\mathcal{K}_{ij}$ is placed in the caches of users i and j. The keys are chosen combinatorially and a general strategy is discussed in Appendix A. The overall cache placement is as follows: $Z_1 = \{A_1, B_1, C_1, \mathcal{K}_{12}, \mathcal{K}_{13}\}$, $Z_2 = \{A_2, B_2, C_2, \mathcal{K}_{12}, \mathcal{K}_{23}\}$ and $Z_3 = \{A_3, B_3, C_3, \mathcal{K}_{13}, \mathcal{K}_{23}\}$. Thus each cache has size $M = 5 \times (1/3) = 5/3$, where $M_D = 1, M_K = 2/3$. Now considering a worst case request where all users request different files, $(d_1, d_2, d_3) = (A, B, C)$, the server can make the transmission, $X_{(A,B,C)} = \{\{A_2 \oplus B_1 \oplus \mathcal{K}_{12}\}, \{A_3 \oplus C_1 \oplus \mathcal{K}_{13}\}, \{B_3 \oplus C_2 \oplus \mathcal{K}_{23}\}\}$, such that everyone can securely retrieve their requested files. Thus $(M, R_s^C) = (5/3, 1)$ is *securely* achievable since $I(X_{(A,B,C)}; A, B, C) = 0$ i.e., a wiretapper would gain no information about the files from the transmission. It can be seen from the cache contents that there are multiple keys in the system thereby avoiding a single point of failure. In general, if we choose operating points (M, R_s^C) such that $M_K > 1/K$, single points of failure in the system can be avoided. Thus the scheme forms an interesting memory-rate trade-off based on users' security constraints which is elaborated subsequently in Remark 1. ◇

Remark 1 (Key Memory vs. Data Memory Trade-Off): The trade-off between the fraction of cache memory occupied by the data and the keys in the secure caching system is shown in Fig. 5 for $N = 5$ files and $K = 5$ users. Consider the cache memory constraint in Theorem 1 i.e., $M \in \frac{N-1}{K}t + 1$, $\forall t \in \{0, 1, 2, \ldots, K\}$. Now, since $M = M_D + M_K$, from Appendix A, we have $M_K = 1 - t/K$ and $M_D = Nt/K$. From Fig. 5, it can be seen that M_K dominates at lower values of M. Formally, $M \geq 2N/(N+1)$, data memory dominates key memory i.e., $M_D > M_K$. From Appendix A, we have $\binom{K}{t+1}$ unique keys in the system. Thus the case for there being only one unique key in the system corresponds to $t = K - 1$ i.e., $M_K = 1/K$. Thus for avoiding one shared key across all users i.e., a single point of failure in the system, we need $M_K > 1/K \Rightarrow t \leq K - 1$, which corresponds to $M \leq (N-1)(K-1)/K + 1$. It is also undesirable that new keys be redistributed to the entire system each time a user leaves. The proposed scheme avoids this scenario by sharing keys. In case a user leaves or is compromised, only the keys contained in that user's cache need to be replaced, leaving the others untouched. Thus, a desirable region of operation would be:

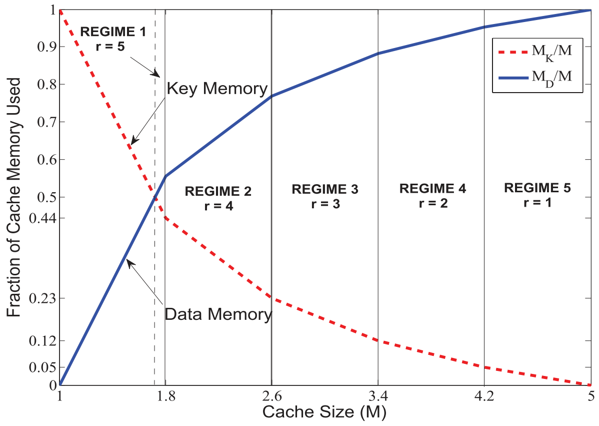


Fig. 5. M_K vs. M_D tradeoff for $N = K = 5$.

$$\frac{2N}{(N+1)} \leq M \leq \frac{(N-1)(K-1)}{K} + 1.$$

In general, a close inspection of Algorithm 1 reveals that when $t > (K - r)$ i.e., when $M > (N-1)(K-r)/K + 1$, a wiretapper can obtain all the keys in the system if it gains access to any r of the K user caches. This means that if r users are compromised, system security will be violated. It is a trivial fact that at $t = 0$, $M = 1$ and each user has one unique key. In this case, the wiretapper will need access to all caches in order to violate the security of the system.

From Fig. 5, we can see that Regime 5, i.e., when $r = 1$, is the weakest regime from the security perspective as there is only one key in the system. Thus operation in Regimes 1–4 is desirable for the case of $N = K = 5$. Now, considering the *conventional secure scheme*, it is seen that there is no sharing of keys as each transmission is useful to only one user. Thus each user stores an unique key of size $|\mathcal{K}| = (1 - \frac{M-1}{N-1})F$ bits. This scheme thus requires the wiretapper to have access to all the caches for the system security to be compromised. Comparing the conventional and proposed schemes from a security perspective, we see that the proposed scheme is a *trade-off* between security and minimization of the rate over the shared link. While the conventional scheme is more difficult to compromise for $M \in \mathbb{N}$, the proposed scheme is able to improve on the transmission rate significantly while still providing security. ◇

B. Intuition Behind Theorem 2 (Converse)

We next present the main idea behind the proof of the converse stated in Theorem 2 through a novel extension of the cut-set bound to incorporate the security constraint. To this end, we focus on the caching system with $N = 2$ files (denoted by A and B) and $K = 2$ users (with cache contents denoted by Z_1 and Z_2). Consider the scenario where user 1 demands file A and user 2 demands file B, i.e., the demand vector is $(d_1, d_2) = (A, B)$. It is easy to check that using the communication $X_{(A,B)}$ from the central server along with the two caches Z_1, Z_2, both files (A, B) can be recovered. This implies the following constraint:

$$H\left(A, B|X_{(A,B)}, Z_1, Z_2\right) \leq \epsilon. \tag{11}$$

Next, for the communication $X_{(A,B)}$ to be secure, we also require the following security constraint to hold:

$$I\left(A, B; X_{(A,B)}\right) \leq \epsilon. \tag{12}$$

Using these two constraints, we next show that for any scheme, $M \geq 1$ must necessarily hold. From the constraints (11)-(12), we have the following sequence of inequalities:

$$\begin{aligned} 2F &\leq H(A, B) = I\left(A, B; X_{(A,B)}, Z_1, Z_2\right) \\ &\qquad + H\left(A, B|X_{(A,B)}, Z_1, Z_2\right) \\ &\overset{(11)}{\leq} I\left(A, B; X_{(A,B)}, Z_1, Z_2\right) + \epsilon \\ &= I\left(A, B; X_{(A,B)}\right) + I\left(A, B; Z_1, Z_2|X_{(A,B)}\right) + \epsilon \\ &\overset{(12)}{\leq} I\left(A, B; Z_1, Z_2|X_{(A,B)}\right) + 2\epsilon \\ &\leq H\left(Z_1, Z_2|X_{(A,B)}\right) + 2\epsilon \leq H(Z_1, Z_2) + 2\epsilon \\ &\leq H(Z_1) + H(Z_2) + 2\epsilon \leq 2MF + 2\epsilon. \end{aligned}$$

This implies

$$M \geq 1 - \frac{\epsilon}{F}. \tag{13}$$

Taking the limit $\epsilon \to 0$, we arrive at the proof of $M \geq 1$. Now consider the fact that given the transmissions from the server $X_{(A,B)}$ for demands $(d_1, d_2) = (A, B)$, $X_{(B,A)}$ for demands $(d_1, d_2) = (B, A)$ and one cache Z_1, both the files A, B can be recovered. Again, we have the following constraints for file retrieval and security:

$$H\left(A, B|X_{(A,B)}, X_{(B,A)}, Z_1\right) \leq \epsilon \tag{14}$$

$$I\left(A, B; X_{(A,B)}\right) \leq \epsilon. \tag{15}$$

Thus we have,

$$\begin{aligned} 2F &\leq H(A, B) = I\left(A, B; X_{(A,B)}, X_{(B,A)}, Z_1\right) \\ &\qquad + H\left(A, B|X_{(A,B)}, X_{(B,A)}, Z_1\right) \\ &\overset{(14)}{\leq} I\left(A, B; X_{(A,B)}, X_{(B,A)}, Z_1\right) + \epsilon \\ &= I\left(A, B; X_{(A,B)}\right) \\ &\qquad + I\left(A, B; X_{(B,A)}, Z_1|X_{(A,B)}\right) + \epsilon \\ &\overset{(15)}{\leq} I\left(A, B; X_{(B,A)}, Z_1|X_{(A,B)}\right) + 2\epsilon \\ &\leq H\left(X_{(B,A)}, Z_1|X_{(A,B)}\right) + 2\epsilon \\ &\leq H\left(X_{(B,A)}\right) + H(Z_1) + 2\epsilon \\ &\leq R_s^* F + MF + 2\epsilon. \end{aligned}$$

This implies that

$$R_s^* + M \geq 2 - \frac{2\epsilon}{F}. \tag{16}$$

Taking the limit $\epsilon \to 0$, we arrive at the proof of $R_s^* + M \geq 2$. We can see that both (13) and (16) hold for all achievable (M, R_s) pairs. Thus we have, $R_s^*(M) \geq 2 - M$ and $M \geq 1$ which gives the lower bound in Fig. 3(b).

IV. Decentralized Caching With Secure Delivery

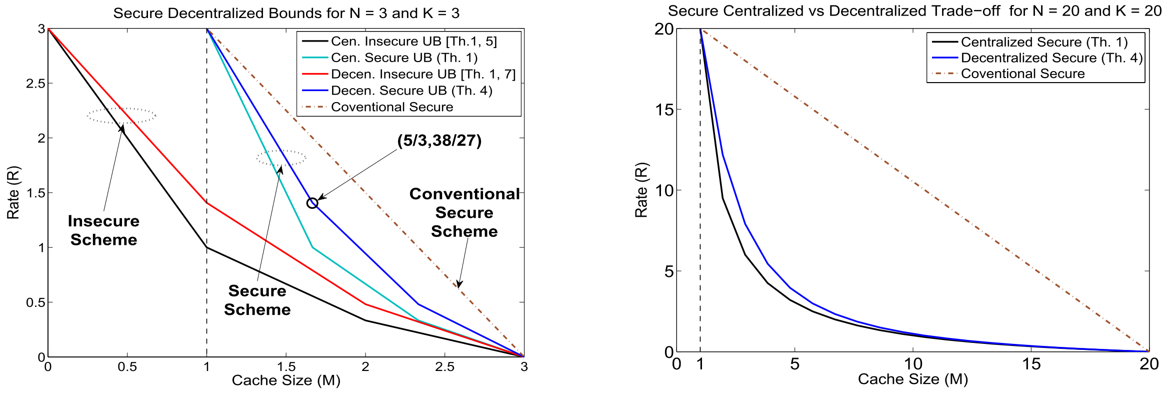
In this section, we extend the secure caching problem to a decentralized caching scheme as discussed in [8]. In the decentralized caching scheme, each user is allowed to cache any random $\frac{M-1}{N-1}$ bits of each of the N files in the system. In the coded delivery scheme, the central server maps the contents of individual users' caches to fragments (which contain non-overlapping combination of bits) in each file. The fragments reflect which user (or set of users) has cached bits contained in the given fragment. This phase is followed by a centralized key placement procedure where the server stores shared keys in each user's cache. The key placement needs to be centralized to maintain key integrity and to secure the files from an external wiretapper. In the delivery phase, the server receives a request $(d_1, \ldots, d_K)$ and forms coded multicast transmissions to extract global caching gain from the system. It then encodes the transmissions with the shared keys and transmits them over the multicast link. The decentralized algorithm is presented in Algorithm 2 in Appendix D. In the case of decentralized caching, similar to the centralized case, the *conventional secure scheme* is one which stores only one unique key per user and exploits only the local caching gain by using encrypted unicast delivery. The transmission rate in this case is given by $K(1 - \frac{M-1}{N-1})$. After the cache placement, the server chooses the scheme which provides the minimum rate over the shared link. The secure rate is then characterized by the following theorem.

Theorem 4: For N files and K users, each with a cache size of $M \in \frac{N-1}{N} \cdot t + 1$, *for* $t \in (0, N]$,

$$\begin{aligned} R_s^D(M) &\triangleq K\left(1 - \frac{M-1}{N-1}\right) \\ &\qquad \cdot \min\left\{\frac{N-1}{K(M-1)} \cdot \left(1 - \left(1 - \frac{M-1}{N-1}\right)^K\right), 1\right\} \end{aligned} \tag{17}$$

is securely achievable. For any $1 < M \leq N$, *the lower convex envelope of these points is achievable.*

The proof of Theorem 4 is given in Appendix D. The variable $t = M_D$, represents the part of the cache memory used to store data at each user (as detailed in Appendix D). Theorem 4 is defined for $t > 0$. At $t = 0$, $M = 1$



(a)

(b)

Fig. 6. (a) (M, R_s^D) trade-off for $N = K = 3$ and (b) Centralized vs. Decentralized Secure Bounds for $N = K = 20$.

Algorithm 2 Secure Decentralized Caching Algorithm

```
Decentralized Cache Placement:
for $k \in \{1, \dots, K\}, n \in \{1, \dots, N\}$ do
    User $k$ randomly caches $\frac{M-1}{N-1}F$ bits of file $n$.
end for
Delivery Procedure for request $(d_1, \dots, d_K)$
Centralized Key Placement:
Central server maps the cache contents to fragments in the files $W_1, \dots, W_N$ and generates keys as follows-
for $i = 0, 1, 2, \dots, K$ do
    for $n = 1, 2, \dots, N$ do
        $W_n = \{W_{n,\mathcal{T}}\}$,  $\mathcal{T} \subseteq \{1, \dots, K\} : |\mathcal{T}| = i$ such that
            $W_{n,\mathcal{T}}$ is cached at user $k$, if $k \in \{\mathcal{T}\}$
    end for
end for
for $s = 1, 2, \dots, K$ do
    for $\mathcal{S} \subseteq \{1, \dots, K\} : |\mathcal{S}| = s$ do
        Key $\mathcal{K}_{\mathcal{S}}$ is generated
        $\mathcal{K}_{\mathcal{S}}$ is placed in cache of user $k$ if $k \in \{\mathcal{S}\}$
    end for
end for
Coded Secure Delivery:
for $s = K, K-1, \dots, 1$ do
    for $\mathcal{S} \subseteq \{1, \dots, K\} : |\mathcal{S}| = s$ do
        Server sends $\left\{\mathcal{K}_{\mathcal{S}} \oplus_{k \in \mathcal{S}} W_{d_k, \mathcal{S} \setminus \{k\}}\right\}$
    end for
end for
Conventional Delivery Procedure for request $(d_1, \dots, d_K)$
Server places individual keys of size $(1 - \frac{M-1}{N-1})F$ bits at
    each user's cache
for $n \in \{0, \dots, N\}$ do
    Server sends enough random linear combinations of bits
        in file $n$ XOR-ed with individual keys for the all users
        requesting it
end for
```

i.e., the caches store a single key of the size of each file. Entire files, XOR-ed with the keys, are then transmitted over the shared link. Thus the rate in this case is $R_s^D(1) \triangleq K$. As before, the same argument holds for the infeasibility of the secure scheme for $M = 0$. The following example illustrates the caching scheme which achieves the rate in Theorem 4.

Example 3: We consider the case for $N = 3$ files and $K = 3$ users, each with a cache of size MF bits. Let the three files be denoted as $(W_1, W_2, W_3) = (A, B, C)$. Fig. 6(a) shows the rate achieved by the secure decentralized caching scheme given by Theorem 4, the rate of the insecure decentralized scheme from [8] and the corresponding centralized bounds. In the decentralized placement phase, each of the 3 users caches a subset of $(M-1)F/2$ bits of each file independently at random. Thus, each bit of a file is cached by a specific user with probability $(M-1)/2$. Considering the file A, the server maps the storage of fragments of file A at the different users' caches into splits, $A_{\mathcal{T}}$, such that $\mathcal{T} \subseteq \{1, 2, 3\}$, $|\mathcal{T}| = i$ for $i = 0, 1, 2, 3$. Thus there are $\sum_{i=0}^{3} \binom{3}{i} = 2^3 = 8$ splits of file A: $(A_\phi, A_1, A_2, A_3, A_{12}, A_{13}, A_{23}, A_{123})$, where A_ϕ consists of bits of A which are not stored in any users' cache. On the other hand, A_{123} has bits which are stored in all users cache. In general, bits in $A_{\mathcal{T}}$ are stored in user k's cache if $k \in \mathcal{T}$. By law of large numbers, we have:

$$|A_{\mathcal{T}}| \approx \left(\frac{M-1}{2}\right)^{|\mathcal{T}|} \left(1 - \frac{M-1}{2}\right)^{3-|\mathcal{T}|} F \text{ bits} \quad (18)$$

with probability approaching one for large enough file size F. The same analysis holds for files B, C. Next, we consider the generation of keys $\mathcal{K}_{\mathcal{S}}$ for $\mathcal{S} \subseteq \{1, 2, 3\}, |\mathcal{S}| = j$ for $j = 1, 2, 3$. Thus the keys generated in the system are: $\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3, \mathcal{K}_{12}, \mathcal{K}_{13}, \mathcal{K}_{23}, \mathcal{K}_{123}$. It can be seen that there are $2^K - 1 = 7$ unique keys in the system. Next we look at the cache contents from the central server's perspective after the centralized key placement phase and before the delivery procedure begins. The cache placement for $N = K = 3$ is given in (19).

$$Z_1 = \left\{ \begin{array}{c} A_1, A_{12}, A_{13}, A_{123} \\ B_1, B_{12}, B_{13}, B_{123} \\ C_1, C_{12}, C_{13}, C_{123} \\ \mathcal{K}_1, \mathcal{K}_{12}, \mathcal{K}_{13}, \mathcal{K}_{123} \end{array} \right\}$$

$$Z_2 = \left\{ \begin{matrix} A_2, A_{12}, A_{23}, A_{123} \\ B_2, B_{12}, B_{23}, B_{123} \\ C_2, C_{12}, C_{23}, C_{123} \\ \mathcal{K}_2, \mathcal{K}_{12}, \mathcal{K}_{23}, \mathcal{K}_{123} \end{matrix} \right\}$$

$$Z_3 = \left\{ \begin{matrix} A_3, A_{13}, A_{23}, A_{123} \\ B_3, B_{13}, B_{23}, B_{123} \\ C_3, C_{13}, C_{23}, C_{123} \\ \mathcal{K}_3, \mathcal{K}_{13}, \mathcal{K}_{23}, \mathcal{K}_{123} \end{matrix} \right\}. \tag{19}$$

The cache placement phase is entirely decentralized as the users do not need to consider the number of other users in the system or their cache contents while storing file fragments in their caches. Next, we consider the delivery procedure of the decentralized caching scheme. The system is characterized based on the worst possible rate over the shared link. Thus we consider a request $(W_{d_1}, W_{d_2}, W_{d_3}) = (A, B, C)$. The server responds by transmitting the reply $X_{(A,B,C)}$. Let the set $\mathcal{S} \subseteq \{1, 2, 3\} : |\mathcal{S}| = s$ for $s = 3, 2, 1$. Then we have $X_{(A,B,C)} = \left\{\mathcal{K}_{\mathcal{S}} \oplus_{k \in \mathcal{S}} W_{d_k, \mathcal{S}\setminus\{k\}} : k = 1, 2, 3\right\}_{s=1}^{3}$, where $W_{d_k,\mathcal{S}\setminus\{k\}}$ corresponds to the fraction of the file W_{d_k}, requested by user k which is not present in user k's cache but is present in the cache of the other $s-1$ users in $\mathcal{S}$. Thus, for $K = 3$ users in the system, the coded secure multicast delivery procedure has 3 phases for each $s = 3, 2, 1$.

For $s = 3$: We have $|\mathcal{S}| = 3 \Rightarrow \mathcal{S} = \{1, 2, 3\}$ and $|\mathcal{S} \setminus \{k\}| = 2$. The transmission is $\{A_{23} \oplus B_{13} \oplus C_{12} \oplus \mathcal{K}_{123}\}$. It can be seen that $\mathcal{K}_{123}$ is associated with sub-files A_{23}, B_{13}, C_{12}. Thus the size of the key is $|\mathcal{K}_{123}| = \max\{|A_{23}|, |B_{13}|, |C_{12}|\}$. In this case, each sub-file is zero padded to the size of the largest sub-file in the set. Considering user 1, we see that Z_1 contains B_{13}, C_{12} and $\mathcal{K}_{123}$. Thus user 1 can XOR out A_{23} from the transmission. It can be seen that the same holds for users 2 and 3. Thus the transmission is useful for all users and the key makes it secure from the wiretapper. For $s = 3$, there is only one transmission of the size of each of these sub-files. Thus, using (18), the rate over the shared link for this transmission is:

$$\left(\frac{M-1}{2}\right)^2 \left(1 - \frac{M-1}{2}\right) F. \tag{20}$$

For $s = 2$: We have $|\mathcal{S}| = 2 \Rightarrow \mathcal{S} \in \{1, 2\}, \{2, 3\}, \{1, 3\}$ and $|\mathcal{S} \setminus \{k\}| = 1$. The transmission for each subset $\mathcal{S}$ is $\{\{A_2 \oplus B_1 \oplus \mathcal{K}_{12}\}, \{B_3 \oplus C_2 \oplus \mathcal{K}_{23}\}, \{A_3 \oplus C_1 \oplus \mathcal{K}_{13}\}\}$.

Again for user 1, we can see that Z_1 contains $B_1, C_1, \mathcal{K}_{12}, \mathcal{K}_{13}$. Thus it can extract A_2, A_3 from this transmission. Similarly the other users can extract fragments of their requested files. In this case, there are three transmissions, each of the size of file fragment, say, A_2. Thus the rate of this transmission is:

$$3 \cdot \left(\frac{M-1}{2}\right) \left(1 - \frac{M-1}{2}\right)^2 F. \tag{21}$$

For $s = 1$: We have $|\mathcal{S}| = 1 \Rightarrow \mathcal{S} \in \{1\}, \{2\}, \{3\}$ and $|\mathcal{S} \setminus \{k\}| = 0$. The transmission for each subset $\mathcal{S}$ is $\left\{\{A_\phi \oplus \mathcal{K}_1\}, \{B_\phi \oplus \mathcal{K}_2\}, \{C_\phi \oplus \mathcal{K}_3\}\right\}$. These transmissions are sent to individual users, containing the residual fragments not stored in each user. The size of each transmission is equal to the size of the file fragments A_ϕ, B_ϕ, C_ϕ. Thus the rate of this transmission is:

$$3 \cdot \left(1 - \frac{M-1}{2}\right)^3 F. \tag{22}$$

Again considering user 1, we can see that the fragments of A not present in its cache i.e., A_ϕ, A_2, A_3, A_{23} are extracted from the entire transmission. The same holds true for the other users. The rate for the composite transmission $X_{(A,B,C)}$ is obtained by summing (20), (21) and (22):

$$\begin{aligned} R_s^D(M)F &= F\left(\frac{M-1}{2}\right)^2 \left(1-\frac{M-1}{2}\right) + 3F\left(\frac{M-1}{2}\right) \\ &\quad \cdot \left(1-\frac{M-1}{2}\right)^2 + 3F\left(1-\frac{M-1}{2}\right)^3 \\ &= 3\left(1-\frac{M-1}{2}\right) \frac{2}{3(M-1)} \left(1-\left(1-\frac{M-1}{2}\right)^3\right) F, \end{aligned} \tag{23}$$

which is the expression given in Theorem 4 for $N = K = 3$. Now, we have $M \in \frac{N-1}{N}\{1, 2, \ldots, N\} + 1 = \left\{\frac{5}{3}, \frac{7}{3}, 3\right\}$. Considering the point $M = 5/3$, we have $R_s^D(M) = 38/27$. Thus the pair $(M, R_s^D) = (5/3, 38/27)$, is securely achievable. This is seen from the (M, R_s^D) trade-off in Fig. 6(a). Similarly other points on the trade-off curve can be evaluated using other feasible values of M. All points on the lines joining the achievable (M, R_s^D) points are also achievable. ◇

Next, we consider the centralized and decentralized trade-off for a large number of files and users. Fig. 6(b) illustrates the case for $N = K = 20$. Compared to Fig. 6(a), we can see that as the number of files and users increase, the decentralized scheme approaches the centralized caching. Thus for large number of files and users, the rates are *asymptotically equal*. This also implies that in the decentralized case, similar to the centralized case, that the cost for security is *almost negligible* when number of files and users increase [24]. The following theorem and corollary compares the rate of the achievable secure decentralized scheme given in Theorem 4 to the lower bound on the rate of the optimal secure scheme given in Theorem 2 and the rate of the achievable secure centralized caching scheme given in Theorem 1.

Theorem 5: Given $R_s^D(M)$ be the rate of the secure decentralized caching scheme given by Algorithm 2 and $R_s^*(M)$ be the rate of the optimal secure caching scheme, for N files and K users, each having a cache size $\frac{N-1}{N} + 1 \leq M \leq N$,

$$\frac{R_s^D(M)}{R_s^*(M)} \leq 17. \tag{24}$$

The proof sketch of Theorem 5 is given in Appendix E. Theorem 5 implies that no scheme, regardless of complexity can improve by more than a constant factor upon the secure decentralized caching scheme presented in Algorithm 2 for the given regime of M. The gap is unbounded only for the case of $K > N$ in the regime $1 \leq M \leq \frac{N-1}{N} + 1$, which is negligibly small for large N, K as discussed in Appendix E.

Corollary 6: Let $R_s^C(M)$ be the rate of the secure centralized caching scheme given in Theorem 1 and $R_s^D(M)$ be the rate of the secure decentralized caching scheme given in Theorem 4. For N files and K users, for $\frac{N-1}{N}+1 \le M \le N$, we have

$$\frac{R_s^D(M)}{R_s^C(M)} \le 17. \tag{25}$$

Corollary 6 is a direct outcome of Theorems 3 and 5. It shows that the decentralized scheme is at most a constant factor 17 worse than the secure centralized scheme in the given regime of M.

V. Discussion and Open Problems

In this section, we discuss some of the open problems and extensions of the current work:

- *Extension to Non-Uniform File Popularities and Multiple Demands per User:* The problem of caching with secure delivery discussed in this paper assumes all files have uniform popularity. We presented an extension of the secure delivery scheme to the case for non-uniform file popularities in [25]. Furthermore, in this paper, we consider the secure caching problem for the case of single requests from users at a given time instant. However, an interesting case is when users demand multiple, say L, files at a given instant. The non-secure problem was addressed from an graph based index coding perspective in [26], while for the secure case, it is an interesting area for future work.
- *Noisy Links & Multiple Eavesdroppers:* In the current treatment of the security problem, it is also interesting to note that the presence of multiple eavesdroppers would not alter the presented results since each eavesdropper would view the same multicast transmission which leaks no information about the files. This is due to the fact that we consider noiseless delivery in this model. The analysis of the problem for multiple eavesdroppers in the presence of noisy links is a direction of future research.
- *Extension to Multiple Requests Over Time:* Another area for future work is the case of security in delivering content for multiple requests over time i.e., security for an online coded caching scheme similar to the one in [10] which would require a key generation technique such that collection of keys over time by an eavesdropper cannot lead to information leakage.
- *Closing the Gap in Small Buffer Case:* Finally closing the gap between the achievable rate and the information theoretic optimal secure rate for $K > N$ in the regime $1 < M < \frac{(K-N)(N-1)}{KN}+1$ for the centralized scheme and $1 < M < \frac{N-1}{N}+1$ for the decentralized scheme, is an interesting open problem.

VI. Conclusion

In this paper, we have analyzed the problem of *secure* caching in the presence of an external wiretapper for both *centralized* and *decentralized* cache placement. We have proposed a key based secure caching strategy which is robust to compromise of users and keys. We have approximated the information theoretic optimal rate of the secure caching problem with novel upper and lower bounds. It has been shown that there is a constant multiplicative gap between the optimal and the achievable rates for the given scheme in case of both centralized and decentralized caching scenarios for most parameters of practical interest. We have shown that for large number of files and users, the secure bounds approach that of the non-secure case i.e., the cost of security in the system is negligible when the number of files and users increase.

Appendix A
Proof of Theorem 1

In this section, we discuss the secure centralized caching strategy which achieves the upper bound $R_s^C(M)$ as stated in Theorem 1. The algorithm achieving the rate in Theorem 1 is presented in Algorithm 1. These are two phases in the caching strategy: the storage phase and the delivery phase. We consider a cache size $M \le N$ and $M \in \frac{N-1}{K}\cdot\{0,1.\dots,K\}+1$. Let $t \in \{0,1,\dots,K\}$ be an integer between 0 and K. The cache memory size can then be parametrized by t as:

$$M = \frac{N-1}{K}t+1 = \frac{Nt}{K}+1-\frac{t}{K}. \tag{26}$$

From (26), we have $t = \frac{K(M-1)}{N-1}$. Next, we break up the total cache memory into data memory and key memory, $M = M_D + M_K$, as follows:

$$M_K = 1-\frac{t}{K}; \quad M_D = M - M_K = \frac{Nt}{K}. \tag{27}$$

From the discussion in Section III, we know that the *conventional secure scheme* achieves the (M, R_s^C) pair $(1, K)$ and $(N, 0)$. Thus $R_s^*(1) \le K$ and $R_s^*(N) = 0$. We therefore consider the case in which $1 < M < N$. In this case, $t \in \{1,2,\dots,K-1\}$.

Storage Phase: In the placement phase, each file W_n for $n = 1,\dots,N$ is split into $\binom{K}{t}$ non-overlapping sub-files of equal size $F/\binom{K}{t}$:

$$W_n = (W_{n,\tau} : \tau \subseteq \{1,\dots,K\}, |\tau| = t). \tag{28}$$

For each n, the sub-file $W_{n,\tau}$ is placed the cache of user k if $k \in \tau$. Since $|\tau| = t$, for each user $k \in \tau$, there are $t-1$ out of $K-1$ possible users with whom it shares a sub-file of a given file W_n. Thus each user caches $N\binom{K-1}{t-1}$ sub-files. Next we generate a set of keys, each of the size of a sub-file i.e. of size $F/\binom{K}{t}$:

$$(\mathcal{K}_{\tau_k} : \tau_k \subseteq \{1,\dots,K\}, |\tau_k| = t+1). \tag{29}$$

The key $\mathcal{K}_{\tau_k}$ is placed in the cache of user k if $k \in \tau_k$. The keys are generated such that all the keys are orthogonal to each other and each key is distributed according to $\mathcal{K}_{\tau_k} \sim \text{unif}\left\{1,2,\dots,2^{F/\binom{K}{t}}\right\}$. Again, since $|\tau_k| = t+1$, each user $k \in \tau_k$ shares key $\mathcal{K}_{\tau_k}$ with t out of $K-1$ possible users. Thus there are $\binom{K-1}{t}$ keys in the cache of each user. Given each key and sub-file has size $F/\binom{K}{t}$, number of bits required for storage at each user is:

$$N\binom{N-1}{t-1}\cdot\frac{F}{\binom{K}{t}}+\binom{K-1}{t}\cdot\frac{F}{\binom{K}{t}}$$
$$=\frac{FNt}{K}+F\left(1-\frac{t}{K}\right)=F\left(\frac{Nt}{k}+1-\frac{t}{K}\right)=FM \qquad (30)$$

which satisfies the memory constraint.

Delivery Phase: We now elaborate on the delivery phase. Consider a request vector $(d_1,\ldots,d_k)\in\{1,\ldots,N^K\}$ where user k requests the file W_{d_k}. Let $\mathcal{S}\subseteq\{1,\ldots,K\}$ be a subset of $|\mathcal{S}|=t+1$ users. Every t users in $\mathcal{S}$ share a sub-file in their cache which is requested by the $t+1$-th user. Given a user $k\in\mathcal{S}$ and $|\mathcal{S}\setminus\{k\}|=t$, the sub-file $W_{d_k,\mathcal{S}\setminus\{k\}}$ is requested by user k as it is a sub-file of W_{d_k} which is missing at user k since $k\notin\mathcal{S}\setminus\{k\}$. The file is present in the cache of the t users $s\in\mathcal{S}\setminus\{k\}$. For each such subset $\mathcal{S}\subseteq\{1,\ldots,K\}$, the server sends the following transmission: $X_{(d_1,\ldots,d_k)}=\left\{\mathcal{K}_{\mathcal{S}}\oplus_{s\in\mathcal{S}}W_{d_s,\mathcal{S}\setminus\{s\}}\right\}$ such that $\{\mathcal{S}\subseteq\{1,2,\ldots,K\}$, $|\mathcal{S}|=t+1\}$. The number of subsets $\mathcal{S}$ is $\binom{K}{t+1}$. Thus there are $\binom{K}{t+1}$ transmissions and an unique key associated with each transmission i.e., there are $\binom{K}{t+1}$ keys in the system. Each transmission has the size of a subfile and thus the total number of bits sent over the rate-limited link is:

$$R_s^C(M)F=\binom{K}{t+1}\cdot\frac{F}{\binom{K}{t}}=\frac{K\left(1-\frac{M-1}{N-1}\right)}{1+\frac{K(M-1)}{N-1}}\cdot F$$
$$\Rightarrow R_s^*(M)\le R_s^C(M)\triangleq\frac{K\left(1-\frac{M-1}{N-1}\right)}{1+\frac{K(M-1)}{N-1}}. \qquad (31)$$

Next, we show that the delivery phase does not reveal any information to the wiretapper i.e., we show that

$$I\left(X_{(d_1,\ldots,d_k)};W_1,\ldots,W_N\right)=0 \qquad (32)$$

We have,

$$\begin{aligned}
&I\left(X_{(d_1,\ldots,d_K)};W_1,\ldots,W_N\right)\\
&\quad=H\left(X_{(d_1,\ldots,d_K)}\right)-H\left(X_{(d_1,\ldots,d_K)}|W_1,\ldots,W_N\right)\\
&\quad=H\left(X_{(d_1,\ldots,d_K)}\right)\\
&\qquad-H\left(\left\{\mathcal{K}_{\mathcal{S}}\oplus_{s\in\mathcal{S}}W_{d_s,\mathcal{S}\setminus\{s\}}:|\mathcal{S}|=t+1\right\}|W_1,\ldots,W_N\right)\\
&\quad=H\left(X_{(d_1,\ldots,d_K)}\right)-H\left(\{\mathcal{K}_{\mathcal{S}}:|\mathcal{S}|=t+1\}|W_1,\ldots,W_N\right)\\
&\quad=H\left(X_{(d_1,\ldots,d_K)}\right)-H\left(\{\mathcal{K}_{\mathcal{S}}:|\mathcal{S}|=t+1\}\right), \qquad (33)
\end{aligned}$$

where, the last equality follows from the fact that the keys are uniformly distributed and are independent of the files $(W_1,\ldots,W_N)$. Using the fact that $H(A,B)\le H(A)+H(B)$, we have:

$$\begin{aligned}
H\left(X_{(d_1,\ldots,d_K)}\right)&=H\left(\left\{\mathcal{K}_{\mathcal{S}}\oplus_{s\in\mathcal{S}}W_{d_s,\mathcal{S}\setminus\{s\}}:|\mathcal{S}|=t+1\right\}\right)\\
&\le\sum_{i=1}^{\binom{K}{t+1}}H\left(\mathcal{K}_{\mathcal{S}_i}\oplus_{s\in\mathcal{S}_i}W_{d_s,\mathcal{S}_i\setminus\{s\}}:|\mathcal{S}_i|=t+1\right)\\
&\le\sum_{i=1}^{\binom{K}{t+1}}\log_2\left(\frac{F}{\binom{K}{t}}\right)=\binom{K}{t+1}\log_2\left(\frac{F}{\binom{K}{t}}\right).
\end{aligned}$$
$$(34)$$

On the other hand, we have:

$$\begin{aligned}
H\left(\{\mathcal{K}_{\mathcal{S}}:|\mathcal{S}|=t+1\}\right)&=\sum_{i=1}^{\binom{K}{t+1}}H\left(\mathcal{K}_{\mathcal{S}_i}:|\mathcal{S}_i|=t+1\right)\\
&=\sum_{i=1}^{\binom{K}{t+1}}\log_2\left(\frac{F}{\binom{K}{t}}\right)=\binom{K}{t+1}\log_2\left(\frac{F}{\binom{K}{t}}\right), \qquad (35)
\end{aligned}$$

where the equality in (35) follows from the fact that the keys $\mathcal{K}_{\mathcal{S}_i}$, for all i are mutually independent and distributed as unif$\left\{1,2,\ldots,2^{F/\binom{K}{t}}\right\}$. Substituting (34) and (35) into (33), we have:

$$I\left(X_{(d_1,\ldots,d_K)};W_1,\ldots,W_N\right)\le 0. \qquad (36)$$

Using the fact that for any X,Y, $I(X;Y)\ge 0$, we have:

$$I\left(X_{(d_1,\ldots,d_K)};W_1,\ldots,W_N\right)=0, \qquad (37)$$

which proves that the rate $R_s^C(M)$ is *securely* achievable. This completes the proof of Theorem 1. □

Appendix B
Proof of Theorem 2

In this section, we prove the information-theoretic lower bound on $R_s^*(M)$ for any $N,K\in\mathbb{N}$. Let s be an integer such that $s\in\{1,\ldots,\min\{N,K\}\}$. Consider the first s caches $Z_1,\ldots,Z_s$. For a request vector $(d_1,d_2,\ldots,d_s,d_{s+1},\ldots,d_K)=(1,2,\ldots,s,\phi,\ldots,\phi)$, the transmission $X_1=X_{(d_1,\ldots,d_k)}$, along with the caches $Z_1,\ldots,Z_s$ must be able to decode the files $W_1,\ldots,W_s$. Similarly there for another request $(d_1,d_2,\ldots,d_s,d_{s+1},\ldots,d_K)=(s+1,s+2,\ldots,2s,\phi,\ldots,\phi)$, the transmission X_2, which along with caches $Z_1,\ldots,Z_s$, must be able to decode the files $W_{s+1},\ldots,W_{2s}$. Thus considering $\lfloor N/s\rfloor$ different requests, the transmissions from the central server denoted by $X_1,\ldots,X_{\lfloor N/s\rfloor}$, along with the caches $Z_1,\ldots,Z_s$, must be able to decode the files $W_1,\ldots,W_{s\lfloor N/s\rfloor}$. Let

$$\begin{aligned}
\widetilde{W}&=\left\{W_1,\ldots,W_{s\lfloor N/s\rfloor}\right\}\\
\widetilde{X}&=\left\{X_1,\ldots,X_{\lfloor N/s\rfloor}\right\}\\
\widetilde{X}_{\setminus\{l\}}&=\left\{X_1,\ldots,X_{l-1},X_{l+1},\ldots,X_{\lfloor N/s\rfloor}\right\}\\
\widetilde{Z}&=\{Z_1,\ldots,Z_s\}.
\end{aligned}$$

In addition, we also have constraints based on file retrieval and security. The file retrieval constraint is based on the fact that given all possible transmissions and caches, all files can can be retrieved. The security constraint is that a wiretapper should not be able to retrieve any information about the files from any transmission by the server. Using Definition 1, we have:

$$H(\widetilde{W}|\widetilde{X},\widetilde{Z})\le\epsilon \qquad (38)$$
$$I(\widetilde{W};X_l)\le\epsilon;\quad l=1,\ldots,\lfloor N/s\rfloor \qquad (39)$$

We present a novel extension to the cut-set bound argument [27] to include the security and file retrieval constraints. Consider the information flow consisting of transmissions $X_1,\ldots,X_{\lfloor N/s\rfloor}$ and caches $Z_1,\ldots,Z_s$ for decoding files $W_1, \dots, W_{s\lfloor N/s \rfloor}$. This flow has minimum capacity $s\lfloor N/s\rfloor$. Thus, we have:

$$
\begin{aligned}
s\lfloor N/s\rfloor F &\le H(\widetilde{W})\\
&= I(\widetilde{W};\widetilde{X},\widetilde{Z}) + H(\widetilde{W}|\widetilde{X},\widetilde{Z}) \overset{(38)}{\le} I(\widetilde{W};\widetilde{X},\widetilde{Z})+\epsilon\\
&= I\left(\widetilde{W};\{X_1,\dots,X_{\lfloor N/s\rfloor}\},\{Z_1,\dots,Z_s\}\right)+\epsilon\\
&= I(\widetilde{W};X_l) + I\left(\widetilde{W};\widetilde{X}_{\backslash\{l\}},\widetilde{Z}|X_l\right)+\epsilon\\
&\overset{(39)}{\le} I\left(\widetilde{W};\widetilde{X}_{\backslash\{l\}},\widetilde{Z}|X_l\right)+2\epsilon\\
&\le H\left(\widetilde{X}_{\backslash\{l\}},\widetilde{Z}\right)+2\epsilon\\
&\le \sum_{i=1,i\neq l}^{\lfloor N/s\rfloor} H(X_i)+\sum_{j=1}^{s}H(Z_j)+2\epsilon\\
&\le (\lfloor N/s\rfloor -1)R_s^*(M)F + sMF+2\epsilon\\
&\Rightarrow s\lfloor N/s\rfloor \le (\lfloor N/s\rfloor-1)R_s^*(M)+sM+\frac{2\epsilon}{F}.
\end{aligned}
\tag{40}
$$

Solving for R_s^* and optimizing over all possible s, we have:

$$
\begin{aligned}
R_s^*(M) &\ge \max_{s\in\{1,\dots,\min\{N,K\}\}} \lim_{\epsilon\to 0} \frac{s\lfloor N/s\rfloor - sM - \frac{2\epsilon}{F}}{\lfloor N/s\rfloor -1}\\
&= \max_{s\in\{1,\dots,\min\{N,K\}\}}\left(s - \frac{s(M-1)}{\left(\lfloor\frac{N}{s}\rfloor-1\right)}\right),
\end{aligned}
\tag{41}
$$

which concludes the proof of Theorem 2. □

Appendix C
Proof of Theorem 3

In this section, we prove that a constant multiplicative gap exists between the securely achievable rate $R_s^C(M)$ given in Theorem 1 and the optimal secure rate $R_s^*(M)$, for the regime

$$\max\left\{\frac{(K-N)(N-1)}{KN}+1, 1\right\} \le M \le N. \tag{42}$$

We consider two cases for the value of K. Firstly, for $K \le N$, we have from Theorem 1:

$$R_s^C(M) \le K\left(1-\frac{M-1}{N-1}\right) = \min\{N,K\}\left(1-\frac{M-1}{N-1}\right). \tag{43}$$

For the case of $K > N$, (42) reduces to $(K-N)(N-1)/KN + 1 \le M \le N$. Thus we have:

$$
\begin{aligned}
&\frac{(K-N)(N-1)}{KN}+1 \le M\\
&\quad\Rightarrow \frac{1}{N}-\frac{1}{K} \le \frac{M-1}{N-1} \quad \Rightarrow K\cdot\frac{1}{1+K\frac{M-1}{N-1}} \le N\\
&\quad\Rightarrow K\left(1-\frac{M-1}{N-1}\right)\frac{1}{1+K\frac{M-1}{N-1}} \le N\left(1-\frac{M-1}{N-1}\right)\\
&\quad\Rightarrow R_s^C(M) \le \min\{N,K\}\left(1-\frac{M-1}{N-1}\right).
\end{aligned}
\tag{44}
$$

To prove the constant gap result, we focus on two cases:

***Case 1** min{N, K} ≤ 17*: Setting $s = 1$ in Theorem 2 gives the following lower bound on the optimal secure rate:

$$R_s^*(M) \ge \left(1-\frac{M-1}{N-1}\right). \tag{45}$$

Hence from (44) and (45), we have

$$\frac{R_s^C(M)}{R_s^*(M)} \le \min\{N,K\} \le 17. \tag{46}$$

***Case 2:** min{N, K} ≥ 18*: For this case, the rate in Theorem 1 has 3 distinct regimes:

- **Regime 1:** $\max\left\{\frac{(K-N)(N-1)}{KN}, 0\right\} \le M-1 \le 1.2\max\left(1,\frac{N-1}{K}\right)$
- **Regime 2:** $1.2\max\left(1,\frac{N-1}{K}\right) < M-1 \le 0.0628(N-1)$
- **Regime 3:** $0.0628(N-1) < M-1 \le N-1$

We consider each of the three regimes separately.

Regime 1: $\max\left\{\frac{(K-N)(N-1)}{KN}, 0\right\} \le M-1 \le 1.2\max\left(1,\frac{N-1}{K}\right)$
By Theorem 1, we have:

$$R_s^C(M) \le R_s^C(1) \le \min\{N,K\}. \tag{47}$$

By Theorem 2 and using the fact that $\lfloor N/s\rfloor \ge N/s - 1$, we have:

$$R_s^*(M) \ge s - \frac{s^2(M-1)}{N-2s}. \tag{48}$$

Setting $s = \lfloor 0.1586\min\{N,K\}\rfloor \in \{1,\dots,\min\{N,K\}\}$ we get, for $M-1 \le 1.2\max\left(1,\frac{N-1}{K}\right)$:

$$
\begin{aligned}
R_s^*(M) &\ge R_s^*\left(1.2\max\left(1,\frac{N-1}{K}\right)+1\right)\\
&\ge 0.1586\min\{N,K\}-1\\
&\quad -\frac{(0.1586\min\{N,K\})^2\cdot 1.2\max\left(1,\frac{N-1}{K}\right)}{N-2\cdot 0.1586\min\{N,K\}}\\
&\ge \min\{N,K\}\left\{0.1586-\frac{1}{\min\{N,K\}}\right.\\
&\quad\left.-\frac{(0.1586)^2\cdot 1.2}{1-2\cdot(0.1586)\min\{1,K/N\}}\right\}\\
&\ge \min\{N,K\}\left\{0.1586-\frac{1}{18}-\frac{1.2\cdot(0.1586)^2}{1-2\cdot 0.1586}\right\}\\
&\ge \frac{1}{17}\min\{N,K\}.
\end{aligned}
\tag{49}
$$

Combining (47) and (49), we have:

$$\frac{R_s^C(M)}{R_s^*(M)} \le 17. \tag{50}$$

Regime 2: $1.2\max\left(1,\frac{N-1}{K}\right) < M-1 \le 0.0628(N-1)$

Let $\bar{M}$ be the largest multiple of $\frac{N-1}{K}$ less than equal to M such that

$$0 \le M-\frac{N-1}{K} \le \bar{M} \le M. \tag{51}$$

Choosing $\bar{M} = M-(N-1)/K$, and using the fact that $R_s^C(M)$ is monotonically decreasing in M, we have:

$$\begin{aligned} R_s^C(M) &\le R_s^C(\bar{M}) \\ &\le K \cdot \left\{1 - \frac{M-1}{N-1} + \frac{1}{K}\right\} \cdot \frac{1}{1 + \frac{K(M-1)}{N-1} - 1} \le \left(\frac{N-1}{M-1}\right), \end{aligned} \tag{52}$$

where we have used $\frac{M-1}{N-1} > \frac{1}{K}$ in the last inequality. Now setting $s = \lfloor 0.1530\frac{N-1}{M-1} \rfloor \in \{1, \dots, \min\{N, K\}\}$ in Theorem 2, we have:

$$\begin{aligned} R_s^*(M) &\ge 0.1530\frac{N-1}{M-1} - 1 - \frac{0.1530^2 \cdot \frac{N-1}{M-1}^2 \cdot (M-1)}{N - 2 \cdot 0.1530 \cdot \frac{N-1}{M-1}} \\ &\ge \frac{N-1}{M-1}\left\{0.1530 - 0.0628 - \frac{0.1530^2}{1 - \frac{2 \cdot 0.1530}{1.2}}\right\} \\ &\ge \frac{1}{17}\left(\frac{N-1}{M-1}\right). \end{aligned} \tag{53}$$

Combining (52) and (53), we get:

$$\frac{R_s^C(M)}{R_s^*(M)} \le 17. \tag{54}$$

Regime 3: $0.0628(N-1) < M - 1 \le N - 1$

Let $\bar{M} - 1$ be a multiple of $(N-1)/K$ less than equal to $0.0628(N-1)$, such that

$$0 \le 0.0628(N-1) - \frac{N-1}{K} \le \bar{M} - 1 \le 0.0628(N-1). \tag{55}$$

Then using Theorem 1 and the fact that $\bar{M} \le M$, we have:

$$\begin{aligned} R_s^C(M) \cdot \frac{1}{1 - \frac{M-1}{N-1}} &\le R_s^C(\bar{M}) \cdot \frac{1}{1 - \frac{\bar{M}-1}{N-1}} \\ \Rightarrow R_s^C(M) &\le R_s^C(\bar{M}) \cdot \frac{1}{1 - \frac{\bar{M}-1}{N-1}} \cdot \left(1 - \frac{M-1}{N-1}\right) \\ &\le R_s^C(\bar{M}) \cdot \frac{1}{1 - 0.0628} \cdot \left(1 - \frac{M-1}{N-1}\right). \end{aligned} \tag{56}$$

Now by Theorem 1 and using (55), we have:

$$R_s^C(\bar{M}) \le \frac{1}{\frac{\bar{M}-1}{N-1} + \frac{1}{K}} \le \frac{1}{0.0628 - \frac{1}{K} + \frac{1}{K}} = \frac{1}{0.0628}. \tag{57}$$

Thus we have, from (56) and (57):

$$R_s^C(M) \le \frac{1}{0.0628(1 - 0.0628)}\left(1 - \frac{M-1}{N-1}\right). \tag{58}$$

Setting $s = 1$ in Theorem 2, we have the following lower bound:

$$R_s^*(M) \ge \left(1 - \frac{M-1}{N-1}\right). \tag{59}$$

Thus combining (58) and (59), we get:

$$\frac{R_s^C(M)}{R_s^*(M)} \le \frac{1}{0.0628(1 - 0.0628)} \le 17. \tag{60}$$

Thus we have proved that for any $N, K \in \mathbb{N}$ and all $\frac{(K-N)(N-1)}{KN} + 1 \le M \le N$, there is a constant multiplicative gap of 17 between the achievable rate and the information theoretic optimal. This concludes the proof of Theorem 3.

Remark 2: For $K \le N$ the gap is bounded for the entire feasible regime of $1 \le M \le N$. However, for $K > N$, the gap is unbounded in the regime:

$$1 \le M < \frac{(K-N)(N-1)}{KN} + 1,$$

and scales with the number of users K. However, $\frac{(K-N)(N-1)}{KN} \le 1$ for any $K > N$ and thus the regime is a fraction of the value of M and is in general negligible when N is large. Also, the regime is always below the values of M for which the data memory dominates key memory i.e., $M > 2N/(N+1) \ge 1$, thereby making it a regime of lesser practical interest.

Appendix D
Proof of Theorem 4

The decentralized algorithm which achieves the rate in Theorem 4 is given in Algorithm 2.

Given N files and K users, each with a cache size of MF bits, we first show that the memory constraint $M \in \frac{N-1}{N}t + 1$ for $t \in (0, N]$ is valid. We then evaluate the rate of Algorithm 2 and show that the multicast delivery is information theoretically secure.

Considering the proposed decentralized scheme in Algorithm 2, each user is allowed to cache any random subset of $\frac{M-1}{N-1}F$ bits of any file W_n. Since the choice of these subsets is uniform, given a particular bit in file W_n, the probability of the bit being cached at a given user is:

$$q \triangleq \frac{M-1}{N-1} \in (0, 1]. \tag{61}$$

Considering a fixed subset of s out of K users, the probability that this bit is cached exactly at these s users and not cached at the remaining $(K-s)$ users is $q^s(1-q)^{K-s}$. The expected number of bits of W_n that are cached at exactly those s users is given by:

$$E\left[\#\text{ of bits of } W_n \text{ at } s \text{ users}\right] = Fq^s(1-q)^{K-s}. \tag{62}$$

The actual realization of the random number of bits of a file W_n cached at s users is within the range:

$$Fq^s(1-q)^{K-s} \pm o(F). \tag{63}$$

For ease of exposition, we consider all the fragments of files shared by s users have the same size. Hence the factor $o(F)$ can be ignored for large enough F.

A. Memory Constraint

Next, the server maps the contents of the users' caches to non-overlapping fragments in files such that each fragment reflects which users have cached the bits contained in the fragment. Referring to Algorithm 2, Line 4, the variable i signifies the number of users which share a given file fragment. For $i = 0$, the file fragments are $W_{n,\phi}$ which is not stored at any user. When $i = 1$, the file fragments are $W_{n,k}$ for $k = 1, \dots, K$ which are stored only at one user and hence shared by none. In general for any i, the fragments $W_{n,\mathcal{S}}$ such that $|\mathcal{S}| = i$ are stored at i users and shared by any given user with $i-1$ other users. Thus, for a given a user k, the number of fragments it shares with $i-1$ out of the remaining $K-1$ users for each i is given by $\binom{K-1}{i-1}$. From (62), we have the size of fragments which are stored at exactly i users is $Fq^i(1-q)^{K-i}$. Thus, the total memory at each user for storing data is given by:

$$M_D F = N \cdot \sum_{i=1}^{K} \binom{K-1}{i-1} Fq^i(1-q)^{K-i}$$

$$M_D = Nq \sum_{i-1=0}^{K-1} \binom{K-1}{i-1} q^{i-1}(1-q)^{(K-1)-(i-1)}$$

$$= Nq = N\frac{M-1}{N-1}. \tag{64}$$

Next, we describe the centralized key placement. For each sub-set $\mathcal{S} \subseteq \{1,\ldots,K\}$ of size s, i.e., $|\mathcal{S}| = s$, where $s = 1, 2, \ldots, K$, a key $\mathcal{K}_\mathcal{S}$ is generated as follows:

$$\mathcal{K}_\mathcal{S} \sim \text{unif}\left\{1, 2, \ldots, 2^{Fq^{s-1}(1-q)^{K-s+1}}\right\}. \tag{65}$$

Subsequently, the key $\mathcal{K}_\mathcal{S}$ is placed in the cache of user k if $k \in \mathcal{S}$. The centralized key generation and placement phase is inherently related to the delivery phase of the decentralized algorithm since the size of a key is related to the size of file fragment which is encoded with the key during coded delivery. Consider the coded delivery phase in Algorithm 2, Line $15-19$. Given a request $(d_1,\ldots,d_K)$, the composite transmission $X_{(d_1,\ldots,d_K)}$ is sent by the server. The composite transmission can be written as:

$$X_{(d_1,\ldots,d_K)} = \left\{X^s_{(d_1,\ldots,d_K)}\right\}_{s=1}^{K}, \tag{66}$$

where $X^s_{(d_1,\ldots,d_K)}$ consists of $\binom{K}{s}$ transmissions, one for each possible sub-set $\mathcal{S}$ of size s i.e.,

$$X^s_{(d_1,\ldots,d_K)} = \left\{\mathcal{K}_\mathcal{S} \oplus_{k\in\mathcal{S}} W_{d_k,\mathcal{S}\setminus\{k\}} : |\mathcal{S}| = s\right\}. \tag{67}$$

$W_{d_k,\mathcal{S}\setminus\{k\}}$ denotes the part of the file W_{d_k} requested by user k which is present in the caches all the users in set $\mathcal{S}$ except in the cache of user k. The key $\mathcal{K}_\mathcal{S}$ is associated with the transmission $\oplus_{k\in\mathcal{S}} W_{d_k,\mathcal{S}\setminus\{k\}}$. Furthermore, from the design of the key placement, the key $\mathcal{K}_\mathcal{S}$ is available in the cache of all the s users in the sub-set $\mathcal{S}$. Since $|\mathcal{S}\setminus\{k\}| = s-1$, from (62) we have, the expected size of the fragment $W_{d_k,\mathcal{S}\setminus\{k\}}$ is given by $Fq^{s-1}(1-q)^{K-s+1}$. For a fixed value of s, the size of each transmission in $X^s_{(d_1,\ldots,d_K)}$ is given by:

$$\max_{k\in\mathcal{S}} |W_{d_k,\mathcal{S}\setminus\{k\}}| = Fq^{s-1}(1-q)^{K-s+1}. \tag{68}$$

Thus, each key $\mathcal{K}_\mathcal{S}$ must be chosen with the size:

$$|\mathcal{K}_\mathcal{S}| = \max_{k\in\mathcal{S}} |W_{d_k,\mathcal{S}\setminus\{k\}}| = Fq^{s-1}(1-q)^{K-s+1}, \tag{69}$$

which is precisely how each key is generated according to (65). Now, for a given value of s, a user k needs file fragments contained in $\mathcal{S}\setminus\{k\}$ i.e., $s-1$ other users in the set $\mathcal{S}$. This set of $s-1$ users need to be chosen out of the remaining $K-1$ users. Thus for each s, there are $\binom{K-1}{s-1}$ keys associated with each user. Thus the total number of keys at each user is given by $\sum_{s=1}^{K}\binom{K-1}{s-1} = 2^{K-1}$. The total memory occupied by keys at each users' cache is given by:

$$M_K F = \sum_{s=1}^{K} \binom{K-1}{s-1} Fq^{s-1}(1-q)^{K-s+1}$$

$$M_K = (1-q)\sum_{s=1}^{K} \binom{K-1}{s-1} Fq^{s-1}(1-q)^{(K-1)-(s-1)}$$

$$= (1-q) = 1 - \frac{M-1}{N-1}. \tag{70}$$

From (70) and (64), we have:

$$M_D + M_K = N\frac{M-1}{N-1} + 1 - \frac{M-1}{N-1} = M, \tag{71}$$

which proves the memory constraint. Putting $M_D = t$, the memory break up can be parametrized as:

$$M = t + (1 - \frac{t}{N}) = \frac{N-1}{N}t + 1. \tag{72}$$

Now, when $t = 0$, $M = 1$, which is the condition for storing just keys in caches and sending entire files over the shared link. On the other hand, when $t = N$, $M = N$ i.e., the entire files are stored in the caches and there is no need for a transmission. Thus $t \in (0, N]$ is the region of interest. Hence $M \in \frac{N-1}{N} \cdot (0, N] + 1$ is valid. Note that the constraint on M is due to the centralized key placement and is thus the cost for security.

Remark 3: Considering the range for file fragment size in (63), if we consider that the fragments are not indeed of equal size, then in turn the key size is also within the range $M_K \pm o(F)$. If this is the case, then the cache memory constraint will be within the range $M \pm o(F)$. Since $o(F)$ can generally be ignored in comparison to M, the cache memory constraint is satisfied on an average. ◇

B. Calculation of $R_s^D(M)$

1) Analysis of Conventional Secure Scheme: In conventional secure delivery scheme, for $N \le K$, the worst case request corresponds to at least one user requesting every file. Considering all users request file W_n, they all have $F(M-1)/(N-1)$ of its bits already in their cache. Thus at most $F\left(1 - \frac{M-1}{N-1}\right) + o(F)$ random linear combinations need to be sent to the users requesting the file n. For ease of exposition, $o(F)$ can be ignored. In the conventional scheme, each user k stores an unique key $\mathcal{K}_k$ of size $\left(1 - \frac{M-1}{N-1}\right)F$ bits which is XOR-ed with the data before transmission. Although there are N files, each users' request needs to be secured with a key. Thus, in contrast to the non-secure case in [8], the unicast delivery is done for K users and the normalized delivery rate is $K\left(1 - \frac{M-1}{N-1}\right)$.

If $N > K$, then at most K different files can be requested. The transmission thus has a normalized rate of $K\left(1 - \frac{M-1}{N-1}\right)$. Thus, for all N and $M \in (1, N]$, the conventional scheme has a normalized rate of:

$$R_s^{\text{conv}}(M) = K\left(1 - \frac{M-1}{N-1}\right) \tag{73}$$

2) Analysis of the Proposed Scheme: Considering the secure delivery procedure for the coded caching scheme in Algorithm 2, we can see that there are $\binom{K}{s}$ subsets $\mathcal{S}$ of cardinality s. Thus there are $\binom{K}{s}$ transmissions for each $s = K, K-1, \ldots, 1$. Now, for the coded secure transmission, the unique key $\mathcal{K}_{\mathcal{S}}$ is associated with each subset $\mathcal{S}$. The total number of unique keys in the system is given by $\sum_{s=1}^{K}\binom{K}{s} = 2^K - 1$.

Now, considering the fragment size of $W_{d_k,\mathcal{S}\setminus\{k\}}$ in (68) and the transmission $X^s_{(d_1,\ldots,d_K)}$ in (67), for each value of s, the size of each transmission is given by:

$$|X^s_{(d_1,\ldots,d_K)}| = \binom{K}{s} F q^{s-1}(1-q)^{K-s+1}. \tag{74}$$

Summing over all values of s, the rate $R^{\text{dec}}_s(M)$, of the composite transmission $X_{(d_1,\ldots,d_K)}$ is:

$$\begin{aligned} R^{\text{dec}}_s(M)F &= \sum_{s=1}^{K}\binom{K}{s} F q^{s-1}(1-q)^{K-s+1} \\ R^{\text{dec}}_s(M) &= \frac{1-q}{q}\cdot\sum_{s=1}^{K}\binom{K}{s} q^{s}(1-q)^{K-s} \\ &= \frac{1-q}{q}\cdot\left(1-(1-q)^K\right) \\ &\overset{(61)}{=} \frac{1-\frac{M-1}{N-1}}{\frac{M-1}{N-1}}\cdot\left(1-\left(1-\frac{M-1}{N-1}\right)^K\right) \\ &= K\left(1-\frac{M-1}{N-1}\right)\cdot\frac{N-1}{K(M-1)} \\ &\quad\cdot\left(1-\left(1-\frac{M-1}{N-1}\right)^K\right). \end{aligned} \tag{75}$$

The server can use either the proposed scheme or the conventional secure scheme, whichever uses the minimal rate. Thus combining (73) and (75), Algorithm 2 achieves a rate of:

$$\begin{aligned} R^D_s(M) &= \min\left\{R^{\text{conv}}_s(M), R^{\text{dec}}_s(M)\right\} \\ &= K\left(1-\frac{M-1}{N-1}\right) \\ &\quad\cdot\min\left\{\frac{N-1}{K(M-1)}\cdot\left(1-\left(1-\frac{M-1}{N-1}\right)^K\right), 1\right\}, \end{aligned} \tag{76}$$

which is the result (17) presented in Theorem 4.

C. Proof of Secure Achievability

Next, we show that the delivery phase does not reveal any information to the wiretapper i.e., we show that:

$$I\left(X_{(d_1,\ldots,d_K)}; W_1,\ldots,W_N\right) = 0 \tag{77}$$

In the decentralized scheme, the central server transmits $X_{(d_1,\ldots,d_K)}$ to satisfy the requests $(d_1,\ldots,d_k)$ of the K users. The composite transmission $X_{(d_1,\ldots,d_K)}$, given in (66), consists of $\binom{K}{s}$ transmissions for each $s = K, K-1, \ldots, 1$. We have:

$$\begin{aligned} &I\left(X_{(d_1,\ldots,d_K)}; W_1,\ldots,W_N\right) \\ &= H\left(X_{(d_1,\ldots,d_K)}\right) - H\left(X_{(d_1,\ldots,d_K)}|W_1,\ldots,W_N\right) \\ &= H\left(X_{(d_1,\ldots,d_K)}\right) - H\left(\left\{X^s_{(d_1,\ldots,d_K)}\right\}_{s=1}^{K}|W_1,\ldots,W_N\right) \\ &= H\left(X_{(d_1,\ldots,d_K)}\right) \\ &\quad -H\left(\left\{\left\{\mathcal{K}_{\mathcal{S}}\oplus_{k\in\mathcal{S}} W_{d_k,\mathcal{S}\setminus\{k\}} : |\mathcal{S}|=s\right\}\right\}_{s=1}^{K}|W_1,\ldots,W_N\right) \\ &= H\left(X_{(d_1,\ldots,d_K)}\right) - H\left(\{\{\mathcal{K}_{\mathcal{S}} : |\mathcal{S}|=s\}\}_{s=1}^{K}|W_1,\ldots,W_N\right) \\ &= H\left(X_{(d_1,\ldots,d_K)}\right) - H\left(\{\{\mathcal{K}_{\mathcal{S}} : |\mathcal{S}|=s\}\}_{s=1}^{K}\right), \end{aligned} \tag{78}$$

where, the last equality follows from the fact that the keys are uniformly distributed and are independent of the files $W_1,\ldots,W_N$. Using the fact that $H(A,B) \le H(A)+H(B)$, we have:

$$\begin{aligned} &H\left(X_{(d_1,\ldots,d_K)}\right) \\ &\quad= H\left(\left\{X^s_{(d_1,\ldots,d_K)}\right\}_{s=1}^{K}\right) \le \sum_{s=1}^{K} H\left(X^s_{(d_1,\ldots,d_K)}\right) \\ &\quad\le \sum_{s=1}^{K}\sum_{i=1}^{\binom{K}{s}} H\left(\mathcal{K}_{\mathcal{S}_i}\oplus_{k\in\mathcal{S}_i} W_{d_k,\mathcal{S}_i\setminus\{k\}} : |\mathcal{S}_i|=s\right) \\ &\quad\le \sum_{s=1}^{K}\sum_{i=1}^{\binom{K}{s}} \log_2\left(Fq^{s-1}(1-q)^{K-s+1}\right) \\ &\quad= \sum_{s=1}^{K}\binom{K}{s}\log_2\left(Fq^{s-1}(1-q)^{K-s+1}\right). \end{aligned} \tag{79}$$

On the other hand, we have:

$$\begin{aligned} &H\left(\{\{\mathcal{K}_{\mathcal{S}} : |\mathcal{S}|=s\}\}_{s=1}^{K}\right) \\ &\quad= \sum_{s=1}^{K} H\left(\{\mathcal{K}_{\mathcal{S}} : |\mathcal{S}|=s\}\right) = \sum_{s=1}^{K}\sum_{i=1}^{\binom{K}{s}} H\left(\mathcal{K}_{\mathcal{S}_i} : |\mathcal{S}_i|=s\right) \\ &\quad= \sum_{s=1}^{K}\sum_{i=1}^{\binom{K}{s}} \log_2\left(Fq^{s-1}(1-q)^{K-s+1}\right) \\ &\quad= \sum_{s=1}^{K}\binom{K}{s}\log_2\left(Fq^{s-1}(1-q)^{K-s+1}\right), \end{aligned} \tag{80}$$

where the equality in (80) follows from the fact that the keys are orthogonal to each other and they are uniformly distributed as in (65). Substituting (79) and (80) into (78), we have:

$$I\left(X_{(d_1,\ldots,d_K)}; W_1,\ldots,W_N\right) \le 0 \tag{81}$$

Using the fact that for any X, Y, $I(X;Y) \ge 0$, we have:

$$I\left(X_{(d_1,\ldots,d_K)}; W_1,\ldots,W_N\right) = 0 \tag{82}$$

which proves that the rate $R^D_s(M)$ is *securely* achievable. This completes the proof of Theorem 4. □

Appendix E
Proof Sketch of Theorem 5

The proof for Theorem 5 is similar to the proof of Theorem 3 in Appendix C. Here, we give a sketch of the proof for completeness. We have to prove that a constant multiplicative gap exists between the achievable decentralized secure rate in Theorem 4 and the information theoretic optimal for the regime:

$$\frac{N-1}{N}+1 \leq M \leq N \tag{83}$$

For the case of $K < N$, from Theorem 4, we have, for $1 < M \leq N$,

$$R_s^D(M) \leq K\left(1-\frac{M-1}{N-1}\right) = \min\{N,K\}\left(1-\frac{M-1}{N-1}\right). \tag{84}$$

Again in the case of $K > N$, we have

$$M \geq \tfrac{N-1}{N}+1 \Rightarrow \frac{N-1}{M-1} < N \tag{85}$$

Now, setting $r = 1-\frac{M-1}{N-1}$ and substituting in (85), we have:

$$\frac{1}{1-r} < N \tag{86}$$

Since $0 \leq r < 1$, we have

$$\frac{1}{1-r} \approx \sum_{i=0}^{K-1} r^i \leq N, \tag{87}$$

which becomes tighter as $K \to \infty$. Noting that (87) is a geometric series, we get:

$$\sum_{i=0}^{K-1} r^i \leq N \quad \Rightarrow \quad \frac{1-r^K}{1-r} \leq N \tag{88}$$

Substituting the value of r, we have:

$$\begin{aligned} &\frac{N-1}{M-1}\left(1-\left(1-\frac{M-1}{N-1}\right)^K\right) \leq N \\ &\Rightarrow R_s^D(M) \leq \min\{N,K\}\left(1-\frac{M-1}{N-1}\right) \end{aligned} \tag{89}$$

Thus in general, $R_s^D(M) \leq \min\{N,K\}\left(1-\frac{M-1}{N-1}\right)$ for the regime:

$$\frac{N-1}{N}+1 \leq M \leq N. \tag{90}$$

The constant multiplicative gap between $R_s^D(M)$ and $R_s^*(M)$ can be proved using ideas similar to Appendix C by considering two cases: $\min\{N,K\} \leq 17$ and $\min\{N,K\} \geq 18$. The proof follows a similar sketch as Appendix C and is detailed in [24]. Again, it is to be noted that for $K > N$, the gap is unbounded in the regime

$$1 < M < \frac{N-1}{N}+1, \tag{91}$$

and scales with the number of users K. But $\frac{N-1}{N} < 1$ for any N and thus the regime of M in which the gap is unbounded is in general negligible, especially when N, K are large. □

References

[1] A. Sengupta, R. Tandon, and T. C. Clancy, "Fundamental limits of caching with secure delivery," in *Proc. Wireless Phys. Layer Secur. Workshop, IEEE Int. Conf. Commun. (ICC)*, Jun. 2014, pp. 771–776.

[2] A. Sengupta, R. Tandon, and T. C. Clancy, "Decentralized caching with secure delivery," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun./Jul. 2014, pp. 41–45.

[3] L. W. Dowdy and D. V. Foster, "Comparative models of the file assignment problem," *ACM Comput. Surv.*, vol. 14, no. 2, pp. 287–313, Jun. 1982.

[4] K. C. Almeroth and M. H. Ammar, "The use of multicast delivery to provide a scalable and interactive video-on-demand service," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 6, pp. 1110–1122, Aug. 1996.

[5] M. R. Korupolu, C. G. Plaxton, and R. Rajaraman, "Placement algorithms for hierarchical cooperative caching," *J. Algorithms*, vol. 38, no. 1, pp. 260–302, 2001.

[6] I. Baev, R. Rajaraman, and C. Swamy, "Approximation algorithms for data placement problems," *SIAM J. Comput.*, vol. 38, no. 4, pp. 1411–1429, Aug. 2008.

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

[8] M. A. Maddah-Ali and U. Niesen, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Trans. Netw.*, to be published.

[9] U. Niesen and M. A. Maddah-Ali, "Coded caching with nonuniform demands," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2014, pp. 221–226.

[10] R. Pedarsani, M. A. Maddah-Ali, and U. Niesen, "Online coded caching," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014, pp. 1878–1883.

[11] U. Niesen and M. A. Maddah-Ali. (Jul. 2014). "Coded caching for delay-sensitive content." [Online]. Available: http://arxiv.org/abs/1407.4489

[12] N. Karamchandani, U. Niesen, M. Maddah-Ali, and S. Diggavi, "Hierarchical coded caching," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun./Jul. 2014, pp. 2142–2146.

[13] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. IEEE INFOCOM*, vol. 1. Mar. 1999, pp. 126–134.

[14] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "On the average performance of caching and coded multicasting with random demands," in *Proc. 11th Int. Symp. Wireless Commun. Syst. (ISWCS)*, 2014, pp. 922–926.

[15] M. Ji, G. Caire, and A. F. Molisch, "Optimal throughput-outage tradeoff in wireless one-hop caching networks," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2013, pp. 1461–1465.

[16] N. Golrezaei, A. G. Dimakis, and A. F. Molisch, "Wireless device-to-device communications with distributed caching," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2012, pp. 2781–2785.

[17] M. Ji, G. Caire, and A. Molisch, "Fundamental limits of distributed caching in D2D wireless networks," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Sep. 2013, pp. 1–5.

[18] M. Ji, G. Caire, and A. F. Molisch. (May 2014). "Fundamental limits of caching in wireless D2D networks." [Online]. Available: http://arxiv.org/abs/1405.5336

[19] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless video content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.

[20] P. Blasco and D. Gündüz, "Learning-based optimization of cache content in a small cell base station," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014, pp. 1897–1903.

[21] P. Blasco and D. Gündüz, "Multi-armed bandit optimization of cache content in wireless infostation networks," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2014, pp. 51–55.

[22] A. Sengupta, S. Amuru, R. Tandon, R. M. Buehrer, and T. C. Clancy, "Learning distributed caching strategies in small cell networks," in *Proc. 11th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Aug. 2014, pp. 917–921.

[23] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, no. 4, pp. 656–715, Sep. 1949.

[24] A. Sengupta, R. Tandon, and T. C. Clancy. (Dec. 2013). "Fundamental limits of caching with secure delivery." [Online]. Available: http://arxiv.org/abs/1312.3961

[25] A. Sengupta, R. Tandon, and T. C. Clancy, "Secure caching with non-uniform demands," in *Proc. IEEE Global Wireless Summit (GWS)*, May 2014, pp. 1–5.

[26] M. Ji, A. M. Tulino, J. Llorca, and G. Caire. (2014). "Order optimal coded delivery and caching: Multiple groupcast index coding." [Online]. Available: http://arxiv.org/abs/1402.4572

[27] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 1991.

Avik Sengupta (S'11) is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA, where he is also with the Hume Center for National Security and Technology. He received the B.Tech. degree in electronics and communication engineering from the West Bengal University of Technology, Kolkata, India, in 2010, and the M.S. degree in electrical and computer engineering from Kansas State University, Manhattan, KS, USA, in 2012. His research interests include wireless communication systems with an emphasis on content distribution and caching in wireless networks, dynamic spectrum access, and optimal resource allocation in LTE networks and cognitive radio systems. He was an Intern with the Advanced Technology Group, Blackberry, Waterloo, ON, Canada, where he worked on LTE system design for D2D communications and with the Wireless Access Laboratory, Huawei Technologies, Markham, ON, where he worked on downlink resource scheduling and power control in multi-eNodeB LTE systems.

Ravi Tandon (S'03–M'09) received the B.Tech. degree in electrical engineering from IIT Kanpur, Kanpur, India, in 2004, and the Ph.D. degree in electrical and computer engineering from the University of Maryland, College Park, MD, USA, in 2010. From 2010 to 2012, he was a Post-Doctoral Research Associate with Princeton University, Princeton, NJ, USA. Since 2012, he has been with the Virginia Polytechnic Institute and State University, Blacksburg, VA, USA, where he is currently a Research Assistant Professor with the Discovery Analytics Center and the Department of Computer Science. His research interests are in the areas of network information theory for wireless networks, information theoretic security, machine learning, and cloud storage systems. He was a recipient of the Best Paper Award at the Communication Theory Symposium at the 2011 IEEE Global Communications Conference.

T. Charles Clancy (S'02–M'06–SM'10) is currently an Associate Professor of Electrical and Computer Engineering with the Virginia Polytechnic Institute and State University (Virginia Tech), Blacksburg, VA, USA, where he is also the Director of the Hume Center for National Security and Technology. Prior to joining Virginia Tech in 2010, he served as a Senior Researcher with the Laboratory for Telecommunications Sciences, a defense research laboratory at the University of Maryland, College Park, MD, USA, where he led research programs in software-defined and cognitive radio. He received the B.S. degree in computer engineering from the Rose-Hulman Institute of Technology, Terre Haute, IN, USA, the M.S. degree in electrical engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, and the Ph.D. degree in computer science from the University of Maryland. He has authored over 100 peer-reviewed technical publications. His current research interests include cognitive communications and spectrum security.