Fundamental Limits of Caching with Secure Delivery

Avik Sengupta, Ravi Tandon, T. Charles Clancy
Hume Center for National Security and Technology,
Department of Electrical and Computer Engineering,
Virginia Tech, Blacksburg, VA USA
Email: {aviksg, tandonr, tcc}@vt.edu

Abstract

Caching is a procedure which allows popular files to be pre-fetched and stored partly in end users’ memory. In modern wireless networks, caching is emerging to play a vital role in reducing peak data rates by storing popular content. In this paper, the concept of information theoretic security for caching is introduced. The proposed secure caching scheme leverages both the local and global caching gains to reduce peak rate while securely delivering requested content to the users. The analysis of the secure caching problem is presented which shows that the proposed scheme introduces security at negligible cost compared to insecure caching schemes, particularly for large number of files and users. It is also shown that the proposed scheme is within a constant multiplicative factor from the information-theoretic optimum for all feasible values of problem parameters.

Index Terms

Caching, Information theoretic security, Secure multicast, Index coding

I. INTRODUCTION

Caching is a technique which helps in reducing the peak network load at times of high traffic volume with the aid of local file storage and content duplication. Fractions of popular files are duplicated and stored apriori in 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 [1]–[4]. In [1], a file assignment viewpoint, analogous to the caching problem, is investigated through different models for assigning files to storage devices. The delivery phase of the caching system can operate as a series of dedicated unicast transmissions to each user by communicating 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 would be multicasting to all users while simultaneously satisfying all demands. Thus, an important aspect in caching is multicast delivery. 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 [3], [4]. Further, their investigations are mainly based on the gains obtained from local content distribution, ignoring the global cache size as a factor for extracting caching gain.

More recently, [5]–[7] have proposed an information theoretic formulation of the caching problem. In [5], 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 multicast delivery phase. This achieves a global caching gain which provides significant improvement over local caching gain. The fundamental concepts presented in [5] are extended in [6], [7] to the more general scenarios of decentralized storage and non-uniform user demands which are modeled by heavy-tailed popularity distributions like the Zipf distribution [8], [9]. Some extensions of the caching problem have also been investigated in the case of Device-to-Device (D2D) communications [9], [10]. The authors in [9] use the interaction of cache memories of the distributed devices to enable efficient frequency reuse.

In this paper, we investigate the fundamental security aspects of the caching problem. 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 an 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 [5], 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 a centralized storage phase, 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 a decentralized storage phase, the user \( k \) is allowed to store any random combination of \( M/N \) bits of each file. User \( k \) (for \( k = 1, \ldots, K \)) then requests access to one of the files \( W_{dk} \) in the database. In the delivery phase, the central server proceeds by transmitting a signal \( X_{(d_1, \ldots, d_K)} \) of size \( RF \) bits over the shared link. Using the content \( Z_k \) (of its cache) and the received signal \( X_{(d_1, \ldots, d_K)} \), the \( k \)-th user intends to reconstruct the requested file \( W_{dk} \). 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 of the \( K \) users to decode its requested file for any set of requests \((d_1, \ldots, d_K)\). 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 [5]–[7].

In this paper, we consider this problem in the presence of an external wiretapper which can observe the multicast communication \( X_{(d_1, \ldots, d_K)} \). This models a scenario in which the communication from the central server to the users occurs over an insecure link. Thus, besides satisfying the users’ demands, we require that the communication, \( X_{(d_1, \ldots, d_K)} \), must not reveal any information about the requested files \((W_{d_1}, \ldots, W_{d_K})\) i.e., \( I(X_{(d_1, \ldots, d_K)}; W_1, \ldots, W_N) = 0 \). 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 [11]. For this problem, a secure 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. Furthermore, the communication over the shared link reveals no information about any file (information theoretically secure). 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 for the secure caching problem. Thus, the function \( R^*_s(M) \) is the fundamental memory-rate tradeoff for the secure caching problem considering uniform user demands and equal file popularity. We investigate both
the centralized cache placement as well as the decentralized placement with secure file delivery.

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 in the system. The system has uniformly distributed orthogonal keys which are stored and shared among 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. It is also shown that the achievable rates of the centralized and decentralized secure caching schemes are asymptotically equal.

II. SYSTEM MODEL

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

$$[2^F] \triangleq \{1, 2, \ldots, 2^F\} \quad (1)$$

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 $(W_1, \ldots, W_N)$ into the cache content:

$$Z_k \triangleq \phi_k(W_1, \ldots, W_N) \quad (2)$$

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 $(W_1, \ldots, W_N)$ to the input

$$X_{(d_1,\ldots,d_K)} \triangleq \psi_{(d_1,\ldots,d_K)}(W_1, \ldots, W_N) \quad (3)$$

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

$$\hat{W}_{(d_1,\ldots,d_K),k} \triangleq \mu_{(d_1,\ldots,d_K),k}(X_{(d_1,\ldots,d_K)}, Z_k) \quad (4)$$

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

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

The information leaked at the wiretapper is defined as:

$$L \triangleq \max_{(d_1,\ldots,d_K) \in [N]^K} I(X_{(d_1,\ldots,d_K)}; W_1, \ldots, W_N) \quad (6)$$

**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

$$R^*_s(M) \triangleq \inf \{ R_s : (M, R_s) \text{ is securely achievable} \}. \quad (7)$$

**Notation:** The following notations are used throughout the rest of the paper: $R^*_s(M)$ denotes the rate achieved by the optimal secure caching scheme as a function of the cache memory size $M$, $R^*_s(M)$ denotes the rate achieved by the centralized caching scheme with secure delivery and $R^*_s(M)$ denotes the rate of the decentralized caching scheme with secure delivery as proposed in this paper. The symbol $\oplus$ denotes the bit-wise XOR operation.

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 caching scheme with secure delivery. We first consider the case where $K \leq N$ i.e., the number of files

$I(X;Y)$ represents the mutual information between two random variables $X$ and $Y$ \cite{12}. 

3
is at least as large as the number of users. 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 $MDF$ bits) and key memory (of size $MKF$ bits) such that $M = MD + MK$. 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 a wiretapper.

**Theorem 1.** For $N$ files and $K \leq N$ users, each with a cache size of $M \in \binom{N-1}{K} \cdot \{0, 1, 2, \ldots, K\} + 1$,

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

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

The proof of Theorem 1 is presented in Appendix A. The centralized caching scheme achieving the rate in Theorem 1 is given in Algorithm 1 in Appendix A. Similar to [5], the achievable rate in (8) consists of three factors. The first factor is $K$. In case of no data caching, this is worst case rate. In order 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 memory size has to be at least $F$ bits to store a key. Thus in the secure problem, $M = 0$ is infeasible and the worst case rate of $K$ is achieved at $M = 1$. Thus in this case $MD = 0$, $MK = 1$ and the $(M, R_s^C)$ pair $(1, K)$ is achievable. Another extreme point is $M = N$ i.e., the case where all files are stored in the user’s cache and no multicasting is required. In this case $MD = N$, $MK = 0$ 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. This scheme naively stores files and keys in the cache and ignores the global caching gain for $1 < M < N$.

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. It is a multicasting gain available to all users for all possible demands. The scheme in Theorem 1 designs the placement of data content and keys in the users’ cache such that coded secure multicasting can be achieved among users for every possible request $(d_1, \ldots, d_K)$. The intelligent placement of keys in the caches of the users and encoding the transmissions during delivery phase introduce security in the system.

We next consider at the case where $K > N$ i.e., the number of users is larger than the number of files in the system.

**Theorem 2.** For $N$ files and $K > N$ users, each with a cache size of $M \in \binom{N-1}{K} \cdot \{0, 1, 2, \ldots, K\} + 1$,

$$R_s^*(M) \leq R_s^C(M) \triangleq K \cdot \left(1 - \frac{M - 1}{N - 1}\right) \min \left\{\frac{1}{1 + K \cdot \frac{M - 1}{N - 1}}, \frac{N}{K}\right\} \tag{9}$$

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

The proof of Theorem 2 is presented in Appendix B. In this regime, the rate is given by the minimum of the rate in Theorem 1 and the rate achieved by the conventional secure scheme. Comparing Theorem 1 to (Th.1, [5]), we can observe that the terms $\frac{M}{N}$ in (Th.1, [5]) 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 (Th.1, [5]) for a given value of $M, N$. This is a cost paid for the security in the system. Next, we characterize the lower bound on $R_s^*(M)$ by the following theorem.

**Theorem 3.** For any $N \in \mathbb{N}$ files and $K \in \mathbb{N}$ users, each having a cache size $1 \leq M \leq N$,

$$R_s^*(M) \geq \max_{s \in \{1, \ldots, \min\{N, K\}\}} \left(\frac{s - s(M - 1)}{\left\lfloor \frac{N}{s} \right\rfloor - 1}\right) \tag{10}$$

The proof of Theorem 3 is presented in Appendix C. We next present a series of examples to explain the intuition.
two caching schemes which achieve the memory rate pairs also achievable. The points on the lines can be achieved by dividing the cache memory and transmitted signal rate on the line joining the two points, we divide each file, of size $F$, into fragments $F/2$ bits. We also generate a key $K_{12}$ which is of the same size as the sub-files i.e., $F/2$ bits. In the storage phase, the server fills the caches as follows: $Z_1 = (A_1, B_1, K_{12})$ and $Z_2 = (A_2, B_2, K_{12})$ 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 $(d_1, d_2) = (A, B)$. From the cache storage phase, it is clear that 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 $X_{(A,B)} = \{A \oplus K_1, B \oplus K_2\}$. This system satisfies every possible request with rate $R = 2$ and it is easily verified that $I(X_{(A,B)}; A, B) = 0$. Thus the $(M, R_s^C)$ pair $(1, 2)$ is securely achievable. Now at the other extreme case when $M = 2$, each user can cache both files and no transmission is necessary. Hence the $(M, R_s^C)$ pair $(2, 0)$ is achievable.

Now we consider the intermediate case in which $M = 3/2$. The scheme for this scenario is depicted in Fig. 2(b). 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 $K_{12} \sim \text{unif}\{1, \ldots, 2^{(F/2)}\}$, 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: $Z_1 = (A_1, B_1, K_{12})$ and $Z_2 = (A_2, B_2, K_{12})$ 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 $(d_1, d_2) = (A, B)$. From the cache storage phase, it is clear that 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 $X_{(A,B)} = \{A_2 \oplus B_1 \oplus K_{12}\}$ which is of rate 1/2. User 1 can obtain $A_2$ by XOR-ing out $B_1, K_{12}$ while user 2 can get $B_1$ by XOR-ing out $A_2, K_{12}$ from $X_{(A,B)}$. A wiretapper, on the other hand, would gain no knowledge of either file from the transmission since $I(X_{(A,B)}; A, B) = 0$ which follows from the fact that the key $K_{12}$ is uniformly distributed. Thus the $(M, R_s^C)$ pair $(3/2, 1/2)$ is securely achievable. This can be seen in the secure upper bound in Fig. 2(a).

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. The points on the lines can be achieved by dividing the cache memory and transmitted signal proportionally between the caching schemes for the two end-points. This can be illustrated as follows. Consider two caching schemes which achieve the memory rate pairs $(M_1, R_1(M_1))$ and $(M_2, R_2(M_2))$. For achieving any rate on the line joining the two points, we divide each file, of size $F$ bits, into fragments $F = F_1 + F_2$ bits where $F_1 = \alpha F$ and $F_2 = (1 - \alpha)F$ bits. The rate achieved by the two schemes are then $R_1(M_1)F_1$ and $R_2(M_2)F_2$ bits. Thus, if these two schemes operate on a shared basis for a cache size $M = M_1 + M_2$, the rate achieved by the
where all users request different files, Thus each cache has size such that everyone can securely retrieve their requested files:

\[ A = (\alpha R_1(M_1) + (1 - \alpha)R_2(M_2),) \]

which gives a point on the line joining the two achievable schemes. Thus, this proves the achievability of the secure upper bound in Fig 2(a). 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, it can be seen that there is only a single key \( K_{12} \) 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. For larger number of users, the system is able to avoid a single point of failure by operating at an appropriate point on secure memory-rate tradeoff. We next illustrate this point through an example.

**Example 2.** We consider the case for \( N = K = 3 \). For this case, from Theorem 1, \( M \in \{1, 5/3, 7/3, 3\} \). Let there be three files \( A, B, C \). The end points \( M = 1 \) and \( M = 3 \) are trivial. We consider the case of \( M = 5/3 \).

The system and bounds for this case is illustrated in Fig. 3(b) and 3(a). 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, \( K_{12}, K_{13}, K_{23} \). In this case each subfile and each key is of size \( F/3 \) bits. In general, the key \( K_{ij} \) 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:

\[
Z_1 = \{A_1, B_1, C_1, K_{12}, K_{13}\}
\]
\[
Z_2 = \{A_2, B_2, C_2, K_{12}, K_{23}\}
\]
\[
Z_3 = \{A_3, B_3, C_3, K_{13}, K_{23}\}.
\]

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 following transmission such that everyone can securely retrieve their requested files:

\[
X_{(A, B, C)} = \left\{ \begin{array}{c} A_2 \oplus B_1 \oplus K_{12} \\ A_3 \oplus C_1 \oplus K_{13} \\ B_3 \oplus C_2 \oplus K_{23} \end{array} \right\}.
\] (11)
The rate for this transmission is $$R_s^C = 3 \times \frac{1}{3} = 1$$. It is easy to see that each user can retrieve its requested file from the transmission and its cache. A wiretapper on the other hand would gain no information about the files $$(A, B, C)$$ from the transmission $$X_{(A,B,C)}$$ since $$I(X_{(A,B,C)}; A, B, C) = 0$$. Thus the $$(M, R_s^C)$$ pair $$(5/3, 1)$$ is securely achievable. It can be seen from the cache contents that there are multiple keys in the system thereby avoiding a single point of failure. If an user/key is compromised, then the keys containing the users’ index can be replaced with new ones or removed. Thus when the $$N = K = 3$$ system operates at this point, it is more robust to key failures. In general, if we choose operating points $$(M, R_s^C)$$ such that $$MK > 1/K$$, single points of failure in the system can be avoided. Thus based on more rigid security constraints, we can choose more keys and a greater transmission rate with lower cache memory. Thus the scheme forms an interesting memory-rate trade-off based on users’ security constraints.

We next present the main idea behind the proof of the converse stated in Theorem 3 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 in which 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:

$$H (A, B|X_{(A,B)}, Z_1, Z_2) \leq \epsilon.$$ (12)

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

$$I (A, B; X_{(A,B)}) \leq \epsilon.$$ (13)

Using these two constraints, we next show that for any scheme, the following bound of Theorem 3 must necessarily hold:

$$M \geq 1.$$ (14)

From the constraints (12)–(13), we have the following sequence of inequalities:

$$2F \leq H(A, B)$$

$$= I (A, B; X_{(A,B)}, Z_1, Z_2) + H (A, B|X_{(A,B)}, Z_1, Z_2)$$

$$\leq I (A, B; X_{(A,B)}, Z_1, Z_2) + \epsilon$$

$$= I (A, B; X_{(A,B)}) + I (A, B; Z_1, Z_2|X_{(A,B)}) + \epsilon$$

$$\leq I (A, B; Z_1, Z_2|X_{(A,B)}) + 2\epsilon$$

$$\leq H (Z_1, Z_2|X_{(A,B)}) + 2\epsilon$$

$$\leq H (Z_1) + H (Z_2) + 2\epsilon$$

$$\leq 2MF + 2\epsilon.$$ (15)

This implies

$$M \geq 1 - \frac{\epsilon}{F}.$$ (16)

Taking the limit $$\epsilon \to 0$$, we arrive at the proof of $$M \geq 1$$.

Now consider the fact that given any two transmissions from the server $$X_{(A,B)}$$, $$X_{(B,A)}$$ and one cache $$Z_1$$, both the files $$A, B$$ can be recovered. Again, we have the following constraints for file retrieval and security:

$$H (A, B|X_{(A,B)}, X_{(B,A)}, Z_1) \leq \epsilon$$ (16)

$$I (A, B; X_{(A,B)}) \leq \epsilon.$$ (17)
Thus we have,
\[
2F \leq H(A, B) \\
= I(A, B; X_{(A,B)}, X_{(B,A)}, Z_1) + H(A, B|X_{(A,B)}, X_{(B,A)}, Z_1) \\
\leq I(A, B; X_{(A,B)}, X_{(B,A)}, Z_1) + \epsilon \\
= I(A, B; X_{(A,B)}) + I(A, B; X_{(B,A)}, Z_1|X_{(A,B)}) + \epsilon \\
\leq I(A, B; X_{(B,A)}, Z_1|X_{(A,B)}) + 2\epsilon \\
\leq H(X_{(B,A)}, Z_1|X_{(A,B)}) + 2\epsilon \\
\leq H(X_{(B,A)}, Z_1) + 2\epsilon \leq H(X_{(B,A)}) + H(Z_1) + 2\epsilon \\
\leq R_s^*F + MF + 2\epsilon. 
\]

This implies that
\[
R_s^* + M \geq 2 - \frac{2\epsilon}{F}. 
\]

Taking the limit \(\epsilon \to 0\), we arrive at the proof of \(R_s^* + M \geq 2\).

We can see that both (15) and (19) 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. 2(a).

Next, we look at the upper bound on the optimal rate for the secure and non-secure schemes for the case when the number of files and users increase. When \(N = K = 20\), it can be seen from Fig. 4(a) that the secure and non-secure bounds almost coincide. For large values of \(N, K\) i.e., \(N = K = 100\), the bounds are practically identical as seen in Fig. 4(b). This result shows that security from a wiretapper can be achieved at almost negligible cost compared to a non-secure scheme.

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., \(M \leq \frac{N-1}{K}t + 1, \forall t \in \{0, 1, 2, \ldots, K\}\). Now, we know that the cache memory is divided into data and key memory \(M = M_D + M_K\). As shown in Appendix A, we have \(M_K = 1 - t/K\) and \(M_D = nt/K\). From Fig. 5, we can see that \(M_K\) dominates at lower values of \(M\). At the point \(M = 2N/(N + 1)\), \(M_D = M_K\). For any \(M \geq 2N/(N + 1)\), data memory dominates. Since the main objective of caching is to reduce traffic at peak network loads, the system should be operated at a trade-off point where the data memory dominates the key memory. Also, in Example 2, we argued that single points of failure in system i.e., a single key for all users can be avoided for
$M_K \geq 1/K$. This corresponds to $t = K - 1$ and $M = (N - 1)(K - 1)/K + 1$. It is to be noted that in a general caching scenario, users may enter or leave the system at any time. In the case that only one key is used across all users, the compromise of the key of a single user leaving the system can violate the security of the entire system. It is also undesirable that new keys be redistributed to the entire system each time an user leaves. The proposed scheme avoids this scenario by sharing keys. In case an user leaves or is compromised, only the keys contained in that user’s cache need to be replaced, leaving the others untouched. Thus we propose the region of operation to be:

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

In general, a close inspection of Algorithm 1 reveals that when $t \leq K - r$ i.e., when $M \geq (N - 1)(K - r)/K + 1$, 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 the users’ caches in order to break down the security of the system. From Fig. 5 we can see that Regime 5 is the weakest from the security perspective as there is only one key in the system. Thus operation in Regimes 1-4 is recommended for the case of $N = K = 5$.

Next, we compare the achievable rate from Theorems 1 and 2 with the lower bound on the optimal rate in Theorem 3 and show that a constant multiplicative gap exists between the optimal $R^*_s(M)$ and the achievable rate.
Theorem 4. For any $N \in \mathbb{N}$ files and $K \in \mathbb{K}$ users, each having a cache size $1 \leq M \leq N$,
\[ 1 \leq \frac{R_s^C(M)}{R_s^S(M)} \leq 17. \tag{20} \]

The proof of Theorem 4 is presented in Appendix [12]. In practice the bound is much tighter as the result in Theorem 4 was obtained using approximations. Using numerical simulations, it is found to be less than 5. In all cases the gap is slightly larger than the gap, 12, obtained in the case of the non-secure caching scheme in [5]. In the next section, we analyse the case of decentralized caching.

IV. DECENTRALIZED CACHING WITH SECURE DELIVERY

In this section, we extend the secure caching problem to a decentralized caching scheme as discussed in [7]. In the decentralized caching scheme, each user $k$ with total cache memory $M_F$ bits, which is once again divided into a data memory ($M_D F$ bits) and key memory ($M_K F$ bits), is allowed to cache any random $\frac{M}{N-1} F$ bits of each of the $N$ files in the system. After the cache placement phase, the central server has two alternatives for delivering the requested files: the coded secure scheme or the conventional secure scheme.

In the coded delivery scheme, the central server aims to extract global caching gain. It 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. Similar to the centralized scheme, the keys are orthogonal to each other and generated uniformly at random. We argue that 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 formally presented in Algorithm 2 in Appendix E.

In the conventional delivery procedure, the server extracts only local caching gain. In this case each user has $\frac{M}{N-1} F$ bits of their requested file already in its cache. The server needs to send the remaining $(1 - \frac{M}{N-1}) F$ bits over the shared link. Considering that each user can only request one file at a time the server needs to store a key of size $(1 - \frac{M}{N-1}) F$ at each user. It can then securely encode and transmit random linear combinations of the order of $\min\{N, K\}(1 - \frac{M}{N-1}) F$ bits [7] for large file size $F$ to satisfy all the requests.

After the cache placement, the central server chooses the scheme which provides the minimum rate over the shared link considering the worst-case request by the users. The secure rate is then characterized by the following theorem.

Theorem 5. For $N$ files and $K$ users, each with a cache size of $M \in \left[\frac{N-1}{N} \cdot t + 1 \right]$ for $t \in (0, N]$,
\[ R_s^D(M) \triangleq K \left(1 - \frac{M-1}{N-1}\right) \min \left\{ \frac{N-1}{K(M-1)} \cdot \left(1 - \frac{M-1}{N-1}\right)^K, \frac{N}{K}\right\} \tag{21} \]
is securely achievable. For any $1 \leq M \leq N$ the lower complex envelope of these points is achievable.

The proof of Theorem 5 is given in Appendix [13]. The variable $t$ represents the part of the cache memory used to store data at each user i.e., $M_D = t$ (as detailed in Appendix [E]). Theorem 5 is defined for $t > 0$. At $t = 0$, we see that $M = 1$ i.e., the caches store a single key of the size of each file and entire files, XOR-ed with the keys, are transferred over the shared link. Thus the rate in this case is $R_s^D(1) \triangleq \min\{N, K\}$. 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 5.

Example 3. We consider the case for $N = 3$ files and $K = 3$ users, each with a cache of size $M$. Let the three files be denoted as $(W_1, W_2, W_3) = (A, B, C)$. Fig. 6 shows the rate achieved by the secure decentralized caching scheme given by Theorem 5 along with the rate of the insecure decentralized scheme from [7] 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 $(M-1)/2$. Theorem 5 was obtained using approximations.
Considering the first file \( A \), the server maps the storage of fragments of file \( A \) at the different users’ caches into splits, \( A_T \), such that \( T \subseteq \{1, 2, 3\} \), \( |T| = i \) for \( i = 0, 1, 2, 3 \). Thus there are \( \sum_{i=0}^{3} \binom{3}{i} = 2^3 = 8 \) splits of file \( A \):

\[
A = (A_\phi, A_1, A_2, A_3, A_{12}, A_{13}, A_{23}, A_{123}),
\]

where \( A_\phi \) 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 \( A_T \) are stored in user \( k \)’s cache if \( k \in T \). By law of large numbers, we have:

\[
|A_T| \approx \left( \frac{M - 1}{2} \right)^{|T|} \left( 1 - \frac{M - 1}{2} \right)^{3-|T|} F
\]

with probability approaching one for large enough file size \( F \). Thus we have:

\[
\begin{align*}
|A_\phi|/F &\approx \left( 1 - \frac{M-1}{2} \right)^3 \\
|A_1|/F &\approx \left( \frac{M-1}{2} \right) \left( 1 - \frac{M-1}{2} \right)^2 \\
|A_{12}|/F &\approx \left( \frac{M-1}{2} \right)^2 \left( 1 - \frac{M-1}{2} \right) \\
|A_{13}|/F &\approx \left( \frac{M-1}{2} \right)^3.
\end{align*}
\]

Note that \( |A_1| \approx |A_2| \approx |A_3| \) and \( |A_{12}| \approx |A_{13}| \approx |A_{23}| \). The same analysis holds for files \( B, C \). Next, we consider the generation of keys \( K_S \) for \( S \subseteq \{1, 2, 3\} \), \( |S| = j \) for \( j = 1, 2, 3 \). Thus the keys generated in the system are:

\[
K_1, K_2, K_3, K_{12}, K_{13}, K_{23}, K_{123}.
\]

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:

\[
Z_1 = \left\{ A_1, A_{12}, A_{13}, A_{123}, B_1, B_{12}, B_{13}, B_{123}, C_1, C_{12}, C_{13}, C_{123}, K_1, K_{12}, K_{13}, K_{123} \right\},
Z_2 = \left\{ A_2, A_{12}, A_{23}, A_{123}, B_2, B_{12}, B_{23}, B_{123}, C_2, C_{12}, C_{23}, C_{123}, K_2, K_{12}, K_{23}, K_{123} \right\},
Z_3 = \left\{ A_3, A_{13}, A_{23}, A_{123}, B_3, B_{13}, B_{23}, B_{123}, C_3, C_{13}, C_{23}, C_{123}, K_3, K_{13}, K_{23}, K_{123} \right\}.
\]
It is to be noted that 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 cache. 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 \(S \subseteq \{1, 2, 3\} : |S| = s\) for \(s = 3, 2, 1\). Then we have:

\[
X_{(A,B,C)} = \{K_S \oplus k \in S W_{d_k \setminus S} : k = 1, 2, 3\} \sum_{s=1}^{3},
\]

where \(W_{d_k \setminus S}\) 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 \(S\). Thus, for \(K = 3\) users in the system, the coded secure multicast delivery procedure has 3 phases for each of \(s = 3, 2, 1\).

For \(s = 3\): We have \(|S| = 3 \Rightarrow S = \{1, 2, 3\}\) and \(|S \setminus \{k\}| = 2\). The transmission is:

\[
\{A_{23} \oplus B_{13} \oplus C_{12} \oplus K_{123}\}.
\]

It can be seen that \(K_{123}\) is associated with sub-files \(A_{23}, B_{13}, C_{12}\). Thus the size of the key is \(|K_{123}| = \max\{|A_{23}|, |B_{13}|, |C_{12}|\}\). In this case, each sub-file is zero padded to the size of the largest sub-file in the set. If we consider each of these files are of equal size, then the key has the size of each sub-file. For ease of exposition, we consider that the sub-files are of approximately equal size.

Considering user 1, we see that \(Z_1\) contains \(B_{13}, C_{12}\) and \(K_{123}\). Thus user 1 can XOR out \(A_{23}\) from the transmission. Thus user 1 can extract the requested part of the information. 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 the rate over the shared link for this transmission is:

\[
\left( \frac{M - 1}{2} \right)^2 \left( 1 - \frac{M - 1}{2} \right) F. \tag{22}
\]

For \(s = 2\): We have \(|S| = 2 \Rightarrow S \in \{1, 2\}, \{2, 3\}, \{1, 3\}\) and \(|S \setminus \{k\}| = 1\). The transmission for each subset \(S\) is:

\[
\{A_2 \oplus B_1 \oplus K_{12}\} \quad \{B_3 \oplus C_2 \oplus K_{23}\} \quad \{A_3 \oplus C_1 \oplus K_{13}\}.
\]

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

\[
3 \cdot \left( \frac{M - 1}{2} \right) \left( 1 - \frac{M - 1}{2} \right)^2 F. \tag{23}
\]

For \(s = 1\): We have \(|S| = 1 \Rightarrow S \in \{1\}, \{2\}, \{3\}\) and \(|S \setminus \{k\}| = 0\). The transmission for each subset \(S\) is:

\[
\{A_1 \oplus K_1\} \quad \{B_1 \oplus K_2\} \quad \{C_1 \oplus K_3\}.
\]

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 \(A_1, B_1, C_1\). Thus the rate of this is:

\[
3 \cdot \left( 1 - \frac{M - 1}{2} \right)^3 F. \tag{24}
\]

Again considering user 1, we can see that the fragments of \(A\) not present in its cache i.e., \(A_1, A_2, A_3, A_{23}\) are extracted from the entire transmission. The same holds true for the other users. The rate for the composite
Secure Centralized vs Decentralized Trade-off for \( N = 20 \) and \( K = 20 \)

Fig. 7. (a) Centralized vs Decentralized Secure Bounds for \( N = K = 20 \) (b) Decentralized Secure vs Non-Secure Bounds for \( N = K = 20 \).

transmission \( X_{(A,B,C)} \) is obtained by summing (22), (23) and (24):

\[
R_s^D(M) F = F \left( \frac{M - 1}{2} \right)^2 \left( 1 - \frac{M - 1}{2} \right) + 3 F \left( \frac{M - 1}{2} \right)^2 \left( 1 - \frac{M - 1}{2} \right)^2 + 3 F \left( 1 - \frac{M - 1}{2} \right)^3
\]

\[
= 3 \left( 1 - \frac{M - 1}{2} \right) \frac{2}{3(M - 1)} \left( 1 - \left( 1 - \frac{M - 1}{2} \right)^3 \right) F,
\]

which is the expression given in Theorem 5 for \( N = K = 3 \). The \((M, R_s^D)\) trade-off for the secure and non-secure schemes for \( N = K = 3 \) is shown in Fig. 6. 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 \((M, R_s^D) = (5/3, 38/27)\) is securely achievable. This is seen from the \((M, R_s^D)\) trade-off in Fig. 6. 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. Compared to the centralized secure scheme, we see that the decentralized scheme performs slightly worse than the centralized i.e., there is a cost incurred due to decentralized data storage. Compared to the non-secure decentralized bound from [7], we also observe that a cost for security is paid i.e., a higher achievable rate for a given cache size.

Next, we consider the centralized and decentralized trade-off for a large number of files and users. Fig. 7(a) illustrates the case for \( N = K = 20 \). Compared to Fig. 6, 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. Comparing with the non-secure decentralized scheme from [7], Fig. 7(b) shows, surprisingly and similar to the centralized case, that the cost for security is almost negligible when number of files and users increase. The following theorem and corollary compares the rate of the achievable secure decentralized scheme given in Theorem 5 to the lower bound on the rate of the optimal secure scheme given in Theorem 3 and the rate of the achievable secure centralized caching scheme given in Theorems 1 and 2.

**Theorem 6.** 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 any \( N \in \mathbb{N} \) files and \( K \in \mathbb{K} \) users, each having a
cache size $1 \leq M \leq N$, 

$$\frac{R^D(M)}{R^*_s(M)} \leq 17.$$  \hfill (26)

The proof of Theorem 6 is given in Appendix F. Theorem 6 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.

**Corollary 7.** Let $R^C_s(M)$ be the rate of the secure centralized caching scheme given in Theorem 1 and $R^D_s(M)$ be the rate of the secure decentralized caching scheme given in Theorem 5. For $N \in \mathbb{N}$ files and $K \in \mathbb{N}$ users, for any $M \in [0, N]$, we have

$$\frac{R^D_s(M)}{R^C_s(M)} \leq 17.$$  \hfill (27)

Corollary 7 is a direct outcome of the results presented in Theorem 4 and Theorem 6. It shows that the decentralized scheme is at most a factor 17 worse than the secure centralized scheme. This is a loose bound and can be numerically tightened to around 1.5.

**V. 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. 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. We also show that for a large number of files and users, the centralized and decentralized secure schemes are asymptotically equal.

**Appendix A**

**Proof of Theorem 1**

In this section, we discuss the secure centralized caching strategy which achieves the upper bound $R^C_s(M)$ as stated in Theorem 1. The algorithm achieving the rate in Theorem 1 is presented in Algorithm 1.

**Algorithm 1 Secure Centralized Caching Algorithm**

**Centralized Cache Placement:** for files $W_1, \ldots, W_N$

1. $t = K(M - 1)/(N - 1)$
2. for $n \in \{1, 2, \ldots, N\}$ do
3. Split file $W_n$ into equal sized fragments $W_{n,T} : T \subseteq \{1, 2, \ldots, K\}, |T| = t$
4. end for
5. Generate keys $K_{T_k}$ such that $T_k \subseteq \{1, 2, \ldots, K\}, |T_k| = t + 1$
6. for $k \in \{1, 2, \ldots, K\}$ do
7. for $n = 1, 2, \ldots, N$ do
8. File $W_{n,T}$ is placed in cache, $Z_k$, of user $k$ if $k \in T$
9. Key $K_{T_k}$ is placed in cache, $Z_k$, of user $k$ if $k \in T_k$
10. end for
11. end for
12. Coded Delivery:
13. for $S$ such that $S \subseteq \{1, 2, \ldots, K\}, |S| = t + 1$ do
14. Server sends $\{K_S \oplus_{k \in S} W_{d_k,S \setminus \{k\}}\}$
15. end for
There are two phases in the caching strategy: the storage phase and the delivery phase. We consider a cache size \( M \leq N \) and \( M \in \frac{N-1}{K} \cdot \{0, 1, \ldots, K\} + 1 \). Let \( t \in \{0, 1, \ldots, K\} \) be an integer between 0 and \( K \). The cache memory size can then be parametrized by \( t \) as:

\[
M = \frac{N - 1}{K} t + 1 = \frac{N t}{K} + 1 - \frac{t}{K}.
\] (28)

From (28), 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:

\[
M_K = 1 - \frac{t}{K}, \quad M_D = M - M_K = \frac{N t}{K}.
\] (29)

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

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

\[
W_n = (W_{n,\tau} : \tau \subseteq \{1, \ldots, K\}, |\tau| = t).
\] (30)

For each \( n \), the sub-file \( W_{n,\tau} \) is placed in 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 \( F/(\binom{K}{t}) \):

\[
(K_{\tau_k} : \tau_k \subseteq \{1, \ldots, K\}, |\tau_k| = t + 1).
\] (31)

The key \( K_{\tau_k} \) 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

\[
K_{\tau_k} \sim \text{unif} \left\{1, 2, \ldots, 2^{ F/(\binom{K}{t})} \right\}.
\] (32)

Again, since \(|\tau_k| = t + 1\), each user \( k \in \tau_k \) shares key \( K_{\tau_k} \) 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:

\[
N \binom{N-1}{t-1} \cdot \frac{F}{\binom{K}{t}} + \binom{K-1}{t} \cdot \frac{F}{\binom{K}{t}} = \frac{F N t}{K} + F \left(1 - \frac{t}{K}\right)
\]

\[
= F \left( \frac{N t}{k} + 1 - \frac{t}{K}\right) = FM,
\] (33)

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 \( S \subseteq \{1, \ldots, K\} \) be a subset of \(|S| = t + 1\) users. Every \( t \) users in \( S \) share a sub-file in their cache which is requested by the \( t + 1 \)-th user. Given a user \( k \in S \) and \(|S \setminus \{k\}| = t\), the sub-file \( W_{d_k, S \setminus \{k\}} \) is requested by user \( k \) as it is a sub-file of \( W_{d_k} \) which is missing at user \( k \) since \( k \notin S \setminus \{k\} \). The file is present in the cache of the \( t \) users \( s \in S \setminus \{k\} \). For each such subset \( S \subseteq \{1, \ldots, K\} \), the server sends the following transmission:

\[
X_{(d_1, \ldots, d_k)} = \left\{K_{S \oplus \tau S} W_{d_k, S \setminus \{s\}} : S \subseteq \{1, 2, \ldots, K\}, |S| = t + 1 \right\}.
\] (34)

The number of subsets \( S \) 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:

\[
RF = \left( \frac{K}{t + 1} \right) \frac{F}{(K_t)} = K - t \cdot F = \frac{K \left( 1 - \frac{M - 1}{N - 1} \right)}{1 + \frac{K(M - 1)}{N - 1}} \cdot F
\]

\[\Rightarrow R_s^*(M) \leq R_s^C(M) = \frac{K \left( 1 - \frac{M - 1}{N - 1} \right)}{1 + \frac{K(M - 1)}{N - 1}}. \quad (35)\]

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

\[I \left( X_{(d_1, \ldots, d_K)}; W_1, \ldots, W_N \right) = 0 \quad (36)\]

We have,

\[
I \left( X_{(d_1, \ldots, d_K)}; W_1, \ldots, W_N \right) = H \left( X_{(d_1, \ldots, d_K)} \right) - H \left( X_{(d_1, \ldots, d_K)} | W_1, \ldots, W_N \right)
\]

\[= H \left( X_{(d_1, \ldots, d_K)} \right) - H \left( \{ \mathcal{K}_S \oplus_{s \in \mathcal{S}} W_{d_s, S \backslash \{s\}} : |S| = t + 1 \} | W_1, \ldots, W_N \right)
\]

\[= H \left( X_{(d_1, \ldots, d_K)} \right) - H \left( \{ \mathcal{K}_S : |S| = t + 1 \} | W_1, \ldots, W_N \right)
\]

\[= H \left( X_{(d_1, \ldots, d_K)} \right) - H \left( \{ \mathcal{K}_S : |S| = t + 1 \} \right), \quad (37)\]

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 \(H(A, B) \leq H(A) + H(B)\) we have:

\[
H \left( X_{(d_1, \ldots, d_K)} \right) \leq \sum_{i=1}^{K \left( t + 1 \right)} H \left( \mathcal{K}_{S_i} \oplus_{s \in \mathcal{S}_i} W_{d_s, S_i \backslash \{s\}} : |S_i| = t + 1 \right)
\]

\[\leq \sum_{i=1}^{K \left( t + 1 \right)} \log_2 \left( \frac{F}{(K_t)} \right) = \left( \frac{K}{t + 1} \right) \log_2 \left( \frac{F}{(K_t)} \right). \quad (38)\]

On the other hand, we have:

\[
H \left( \{ \mathcal{K}_S : |S| = t + 1 \} \right) = \sum_{i=1}^{K \left( t + 1 \right)} H \left( \mathcal{K}_{S_i} : |S_i| = t + 1 \right) = \sum_{i=1}^{K \left( t + 1 \right)} \log_2 \left( \frac{F}{(K_t)} \right) = \left( \frac{K}{t + 1} \right) \log_2 \left( \frac{F}{(K_t)} \right). \quad (39)\]

where the equality in (39) follows from the fact that the keys are orthogonal to each other and they are uniformly distributed as in (32). Substituting (38) and (39) into (37), we have:

\[I \left( X_{(d_1, \ldots, d_K)}; W_1, \ldots, W_N \right) \leq 0. \quad (40)\]

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

\[I \left( X_{(d_1, \ldots, d_K)}; W_1, \ldots, W_N \right) = 0, \quad (41)\]

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

**APPENDIX B**

**PROOF OF THEOREM 2**

In this section, we characterize the achievable rate for the case of \(K > N\) i.e., the number of users is greater than the number of files. In this case, there is a natural opportunity of multicasting all \(N\) files to \(K\) users. In our

\(^2\)Equality holds only if \(A\) and \(B\) are orthogonal (independent) to each other \(^3\).
secure setting, a conventional caching scheme [5] achieves a rate of
\[ R(M) = \min\{N, K\} \left( 1 - \frac{MD}{N} \right) \]
\[ = \min\{N, K\} \left( 1 - \frac{t}{K} \right) \]
\[ = \min\{N, K\} \left( 1 - \frac{M - 1}{N - 1} \right). \quad (42) \]

The rate obtained in Theorem 2 is a minimum of the rate in Theorem 1 and \( R(M) \):
\[ R^C_s(M) = \min \left\{ \frac{K \left( 1 - \frac{M - 1}{N - 1} \right)}{1 + \frac{K(M - 1)}{N - 1}}, \min\{N, K\} \left( 1 - \frac{M - 1}{N - 1} \right) \right\}. \quad (43) \]

For \( K > N \), (43) reduces to:
\[ R^C_s(M) = K \cdot \left( 1 - \frac{M - 1}{N - 1} \right) \min \left\{ \frac{1}{1 + \frac{K(M - 1)}{N - 1}}, \min\left( \frac{N}{K}, 1 \right) \right\}. \quad (44) \]

This concludes the proof of Theorem 2.

**APPENDIX C**

**PROOF OF THEOREM 3**

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 \( X_1 = X_{(d_1, \ldots, d_s)} \), 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
\[ \tilde{W} = \{ W_1, \ldots, W_s \lfloor N/s \rfloor \} \]
\[ \tilde{X} = \{ X_1, \ldots, X_{\lfloor N/s \rfloor} \} \]
\[ \tilde{X}_{\{t\}} = \{ X_1, \ldots, X_{t-1}, X_{t+1}, \ldots, X_{\lfloor N/s \rfloor} \} \]
\[ \tilde{Z} = \{ Z_1, \ldots, Z_s \}. \]

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 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:
\[ H(\tilde{W} | \tilde{X}, \tilde{Z}) \leq \epsilon \quad (45) \]
\[ I(\tilde{W}; X_l) \leq \epsilon; \quad l = 1, \ldots, \lfloor N/s \rfloor \quad (46) \]

We present a novel extension to the cut-set bound argument [12] to include the security and file retrieval.
Thus we have:

\[ s \lfloor \frac{N}{s} \rfloor F \leq H(\tilde{W}) \]

\[ = I(\tilde{W}; \tilde{X}, \tilde{Z}) + H(\tilde{W} | \tilde{X}, \tilde{Z}) \]

\[ \leq I(\tilde{W}; \tilde{X}, \tilde{Z}) + \epsilon \]

\[ = I\left(\tilde{W}; \{X_1, \ldots, X_{\lfloor N/s \rfloor}\}, \{Z_1, \ldots, Z_s\}\right) + \epsilon \]

\[ = I(\tilde{W}; X_i) + I\left(\tilde{W}; \tilde{X}_{\lfloor \ell \rfloor}, \tilde{Z} | X_i\right) + \epsilon \]

\[ \leq I\left(\tilde{W}; \tilde{X}_{\lfloor \ell \rfloor}, \tilde{Z} | X_i\right) + 2\epsilon \]

\[ \leq H\left(\tilde{X}_{\lfloor \ell \rfloor}, \tilde{Z}\right) + 2\epsilon \]

\[ \leq \sum_{i=1, \ell \neq i}^{\lfloor N/s \rfloor} H(X_i) + \sum_{j=1}^{s} H(Z_j) + 2\epsilon \]

\[ \leq (\lfloor N/s \rfloor - 1) R_s^*(M) F + s MF + 2\epsilon \]

\[ \Rightarrow s \lfloor \frac{N}{s} \rfloor \leq (\lfloor N/s \rfloor - 1) R_s^*(M) + s M + \frac{2\epsilon}{F}. \quad (47) \]

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

\[ R_s^*(M) \geq \max_{s \in \{1, \ldots, \min\{N, K\}\}} \lim_{\ell \to 0} \frac{s \lfloor N/s \rfloor - sM - 2\epsilon}{\lfloor N/s \rfloor - 1} \]

\[ = \max_{s \in \{1, \ldots, \min\{N, K\}\}} \left( s - s \frac{(M - 1)}{\left(\frac{N}{s} - 1\right)} \right), \quad (48) \]

which concludes the proof of Theorem 3. \( \square \)

---

**APPENDIX D**

**PROOF OF THEOREM 4**

In this section, we prove that a constant multiplicative gap exists between the securely achievable rate \( R_s^C(M) \) given in Theorems 1 and 2 and the optimal secure rate \( R_s^*(M) \). We will consider two cases: \( \min\{N, K\} \leq 17 \) and \( \min\{N, K\} \geq 18 \). For \( \min\{N, K\} \leq 17 \), we have from Theorem 2:

\[ R_s^C(M) \leq \min\{N, K\} \left(1 - \frac{M - 1}{N - 1}\right). \quad (49) \]

Also, setting \( s = 1 \) in Theorem 3 gives:

\[ R_s^*(M) \geq \left(1 - \frac{M - 1}{N - 1}\right). \quad (50) \]

Thus we have:

\[ \frac{R_s^C(M)}{R_s^*(M)} \leq \min\{N, K\} \leq 17 \quad (51) \]

For \( \min\{N, K\} \geq 18 \), we consider the fact that the rate in Theorem 2 has 3 distinct regimes:

**Regime 1:** \( 0 \leq M - 1 \leq 1.2\max\left(1, \frac{N-1}{K}\right) \) where \( R_s^C(M) \) is linear

**Regime 2:** \( 1.2\max\left(1, \frac{N-1}{K}\right) < M - 1 \leq 0.0628(N - 1) \) where \( R_s^C(M) \) is non-linear

**Regime 3:** \( 0.0628(N - 1) < M - 1 \leq N - 1 \) where \( R_s^C(M) \) is once again linear

---

18
We consider each of the three regimes separately.

**Regime 1:** $0 \leq M - 1 \leq 1.2 \max \left(1, \frac{N-1}{K}\right)$

By Theorem 3, we have:

$$R_s^C(M) \leq R_s^C(1) \leq \min\{N, K\}. \quad (52)$$

By Theorem 5 and using the fact that $\lfloor N/s \rfloor \geq N/s - 1$, we have:

$$R_s^*(M) \geq s - \frac{s^2(M - 1)}{N - 2s}. \quad (53)$$

Setting $s = \lceil 0.1586 \min\{N, K\} \rceil \in \{1, \ldots, \min\{N, K\}\}$ we get, for $M - 1 \leq 1.2 \max \left(1, \frac{N-1}{K}\right)$:

$$R_s^*(M) \geq R_s^* \left(1.2 \max \left(1, \frac{N-1}{K}\right) + 1\right) \geq 0.1586 \min\{N, K\} - 1 - \frac{(0.1586 \min\{N, K\})^2 \cdot 1.2 \max \left(1, \frac{N-1}{K}\right)}{N - 2 \cdot 0.1586 \min\{N, K\}} \geq \min\{N, K\} \left\{0.1586 - \frac{1}{18} \cdot \frac{(0.1586)^2}{1 - 2 \cdot 0.1586} \right\} \geq \frac{1}{17} \min\{N, K\}. \quad (54)$$

Combining (52) and (54), we have:

$$\frac{R_s^C(M)}{R_s^*(M)} \leq 17. \quad (55)$$

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

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

$$0 \leq M - \frac{N-1}{K} \leq \bar{M} \leq M. \quad (56)$$

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

$$R_s^C(M) \leq R_s^C(\bar{M}) \leq K \cdot \left\{1 - \frac{M - 1}{N - 1} + \frac{1}{K}\right\} \cdot \frac{1}{1 + \frac{K(M-1)}{N-1} - 1} \leq \left(\frac{N-1}{M-1}\right), \quad (57)$$

where we have used $\frac{N-1}{N-1} > \frac{1}{K}$ in the last inequality. Now setting $s = \lceil 0.1530 \frac{N-1}{M-1} \rceil \in \{1, \ldots, \min\{N, K\}\}$ in Theorem 3, we have:

$$R_s^*(M) \geq 0.1530 \frac{N-1}{M-1} - 1 - \frac{0.1530^2 \cdot \frac{N-1}{M-1}^2 \cdot (M-1)}{N - 2 \cdot 0.1530 \cdot \frac{N-1}{M-1}} \geq \frac{N-1}{M-1} \left\{0.1530 - 0.0628 - \frac{0.1530^2}{1 - 2 \cdot 0.1530}\right\} \geq \frac{1}{17} \left(\frac{N-1}{M-1}\right). \quad (58)$$

Combining (57) and (58), we get:

$$\frac{R_s^C(M)}{R_s^*(M)} \leq 17. \quad (59)$$
Regime 3: \(0.0628(N - 1) < M - 1 \leq N - 1\)

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

\[
0 \leq 0.0628(N - 1) - \frac{N - 1}{K} \leq M - 1 \leq 0.0628(N - 1).
\] (60)

Then we have:

\[
R^C_s(M) \cdot \frac{1}{1 - \frac{M - 1}{N - 1}} \leq R^C_s(M) \cdot \frac{1}{1 - \frac{M - 1}{N - 1}} \\
\leq R^C_s(M) \cdot \frac{1}{1 - \frac{M - 1}{N - 1}} \cdot \left(1 - \frac{M - 1}{N - 1}\right)
\leq R^C_s(M) \cdot \frac{1}{1 - 0.0628} \cdot \left(1 - \frac{M - 1}{N - 1}\right).
\] (61)

Now by Theorem 1 and using (60), we have:

\[
R^C_s(M) \leq \frac{1}{\bar{M} - 1} - \frac{1}{K}
\leq \frac{1}{0.0628 - \frac{1}{K}} + \frac{1}{K}
= \frac{1}{0.0628}.
\] (62)

Thus we have, from (61) and (62):

\[
R^C_s(M) \leq \frac{1}{0.0628(1 - 0.0628)} \left(1 - \frac{M - 1}{N - 1}\right).
\] (63)

Setting \(s = 1\) in Theorem 3 we have again:

\[
R^*_s(M) \geq \left(1 - \frac{M - 1}{N - 1}\right).
\] (64)

Thus combining (63) and (64), we get:

\[
\frac{R^C_s(M)}{R^*_s(M)} \leq \frac{1}{0.0628(1 - 0.0628)} \leq 17.
\] (65)

Thus we have proved that for any \(N, K \in \mathbb{N}\) and all \(1 \leq M \leq N\), there is a constant multiplicative gap of 17 between the achievable rate and the information theoretic optimal. This concludes the proof of Theorem 4.

Appendix E

Proof of Theorem 5

The decentralized algorithm which achieves the rate in Theorem 5 is given in Algorithm 2. Given \(N\) files and \(K\) users, each with a cache size of \(MF\) bits, we first aim to show that the memory constraint \(M \in \frac{N-1}{2} t + 1\) for \(t \in (0, N]\) is valid. We will 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:

\[
q \triangleq \frac{M - 1}{N - 1} \in (0, 1].
\] (66)

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
Algorithm 2 Secure Decentralized Caching Algorithm

Decentralized Cache Placement:
1: for $k \in \{1, \ldots, K\}, n \in \{1, \ldots, N\}$ do
2: User $k$ caches a random subset of $\frac{M-1}N F$ bits of file $n$.
3: end for

Delivery Procedure for request $(d_1, \ldots, d_K)$

Centralized Key Placement:
Central server maps the cache contents fragments in the files $W_1, \ldots, W_N$ and generates keys as follows-
4: for $i = 0, 1, 2, \ldots, K$ do
5: for $n = 1, 2, \ldots, N$ do
6: $W_n = \{W_n, T\}$, $T \subseteq \{1, \ldots, K\}$ such that $W_n, T$ is cached at user $k$, if $k \in \{T\}$
7: end for
8: end for
9: for $s = 1, 2, \ldots, K$ do
10: for $S \subseteq \{1, \ldots, K\}$ : $|S| = s$ do
11: Key $K_{\langle S \rangle}$ is generated
12: $K_{\langle S \rangle}$ is placed in cache of user $k$ if $k \in \{S\}$
13: end for
14: end for

Coded Secure Delivery:
15: for $s = K, K-1, \ldots, 1$ do
16: for $S \subseteq \{1, \ldots, K\}$ : $|S| = s$ do
17: Server sends $\{K_{\langle S \rangle} \oplus_{k \in S} W_{d_k, S\{\langle k\rangle\}}\}$
18: end for
19: end for

Conventional Delivery Procedure for request $(d_1, \ldots, d_K)$
20: Server places individual keys of size $(1 - \frac{M-1}N)F$ bits at each user’s cache
21: for $n \in \{0, \ldots, N\}$ do
22: Server sends enough random linear combinations of bits in file $n$ XOR-ed with individual keys for the all users requesting it
23: end for

at exactly those $s$ users is given by:

$$E[\#\text{ of bits of } W_n \text{ cached at } s \text{ users and not cached at } (K-s) \text{ users}] = Fq^s(1-q)^{K-s}. \quad (67)$$

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

$$Fq^s(1-q)^{K-s} \pm o(F). \quad (68)$$

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 $F$.

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. First, we aim to calculate the number of fragment of each file that a user stores. Referring to Algorithm 2 Line 5, 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 $k = 1, \ldots, K$ which are stored only at one user and hence shared by none. In general for any $i$, the fragments $W_{n,S}$ such that $|S| = i$ 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 \((67)\), we have the size of fragments which are stored at exactly \( i \) users is \( F q^i(1 - q)^{K - i} \). Thus, the total memory at each user for storing data is given by:

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

\[
M_D = N q \sum_{i=1}^{K-1} \binom{K - 1}{i - 1} q^{i-1}(1 - q)^{(K-1)-(i-1)}
= N q = N^{M - 1} \frac{1}{N - 1}.
\]  

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

\[
K_S \sim \text{unif}\left\{1, 2, \ldots, 2F q^{s-1}(1 - q)^{K-s+1}\right\}.
\]  

Subsequently, the key \( K_S \) is placed in the cache of user \( k \) if \( k \in S \). 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 central server. The composite transmission can thus be written as:

\[
X(d_1, \ldots, d_K) = \left\{X^{s}_{(d_1, \ldots, d_K)}\right\}_{s=1}^{K},
\]

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

\[
X^{s}_{(d_1, \ldots, d_K)} = \left\{K_S \oplus_{k \in S} W_{d_k, S \setminus \{k\}} : |S| = s \right\}.
\]

\( W_{d_k, S \setminus \{k\}} \) denotes the part of the file \( W_{d_k} \) requested by user \( k \) which is present in the caches all the users in set \( S \) except the cache user \( k \). The key \( K_S \) is associated with the transmission \( \oplus_{k \in S} W_{d_k, S \setminus \{k\}} \). Furthermore, from the design of the key placement, the key \( K_S \) is available in the cache of all the \( s \) users in the sub-set \( S \).

Since \( |S \setminus \{k\}| = s - 1 \), from \((67)\) we have, the expected size of the fragment \( W_{d_k, S \setminus \{k\}} \) is given by:

\[
F q^{s-1}(1 - q)^{K-s+1}.
\]

For a fixed value of \( s \), the size of each transmission in \( X^{s}_{(d_1, \ldots, d_K)} \) is given by:

\[
\max_{k \in S} |W_{d_k, S \setminus \{k\}}| = F q^{s-1}(1 - q)^{K-s+1}.
\]

Thus, each key \( K_S \) must be chosen with the size:

\[
|K_S| = \max_{k \in S} |W_{d_k, S \setminus \{k\}}| = F q^{s-1}(1 - q)^{K-s+1},
\]

which is precisely how each key is generated according to \((70)\).

Now, for a given value of \( s \), a user \( k \) needs file fragments contained in \( S \setminus \{k\} \) i.e., \( s - 1 \) other users in the set \( S \). 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:

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

\[ M_K = (1-q) \sum_{s=1}^{K} \binom{K-1}{s-1} F q^{s-1} (1-q)^{(K-1)-(s-1)} = (1-q) = 1 - \frac{M-1}{N-1}. \]  \( (76) \)

From \( (76) \) and \( (69) \), we have:

\[ M_D + M_K = N \frac{M-1}{N-1} + 1 - \frac{M-1}{N-1} = M, \]  \( (77) \)

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

\[ M = t + (1 - \frac{t}{N}) = \frac{N-1}{N} t + 1. \]  \( (78) \)

Now, when the data memory \( t = 0 \), \( M = 1 \) 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 1.** Considering the range for file fragment size in \( (68) \), 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.

**Calculation of \( R^D_s(M) \)**

**A. Analysis of Conventional Secure Scheme**

We begin with the analysis of the conventional secure delivery scheme. For \( N \leq K \), the worst case request corresponds 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 for \( F \) large enough, 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 \( K_k \) of size \( \left( 1 - \frac{M-1}{N-1} \right) F \) bits which is XOR-ed with the data before transmission. As this is done for all \( N \) files, the normalized delivery rate is:

\[ N \left( 1 - \frac{M-1}{N-1} \right). \]

If \( N > K \), then at most \( K \) different files can be requested. The transmission after XOR-ing with a key \( K_k \) for each user \( k \) has a normalized rate of:

\[ K \left( 1 - \frac{M-1}{N-1} \right). \]

Thus for all \( N \) and \( M \in (0, N] \), the conventional scheme has a normalized rate of:

\[ R(M) = \min\{N, K\} \left( 1 - \frac{M-1}{N-1} \right) = K \left( 1 - \frac{M-1}{N-1} \right) \min\{1, N/K\}. \]  \( (79) \)

**B. 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 \( S \) 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 $K_S$ is associated with each subset $S$. The total number of unique keys in the system is given by:

$$\sum_{s=1}^{K} \binom{K}{s} = 2^K - 1. \quad (80)$$

Now, considering the fragment size of $W_{d_{k}, S \setminus \{k\}}$ in (73) and the transmission $X_s^{(d_1, \ldots, d_K)}$ in (72), for each value of $s$, the size of each transmission is given by:

$$|X_s^{(d_1, \ldots, d_K)}| = \binom{K}{s} F_q^{s-1}(1-q)^{K-s+1}. \quad (81)$$

Summing over all values of $s$, the rate of the composite transmission $X_{(d_1, \ldots, d_K)}$ is:

$$R(M)F = \sum_{s=1}^{K} \binom{K}{s} F_q^{s-1}(1-q)^{K-s+1}$$

$$= \frac{1-q}{q} \cdot \sum_{s=1}^{K} q^s(1-q)^{K-s}$$

$$= \frac{1-q}{q} \cdot \left( \sum_{s=0}^{K} q^s(1-q)^{K-s} - (1-q)^{K} \right)$$

$$= \frac{1-q}{q} \cdot (1 - (1-q)^{K})$$

$$= \frac{1-M^{-1}}{N-1} \cdot \left( 1 - \left( 1- \frac{M-1}{N-1} \right)^K \right)$$

$$= K \left( 1 - \frac{M-1}{N-1} \right) \cdot \frac{N-1}{K(M-1)} \cdot \left( 1 - \left( 1- \frac{M-1}{N-1} \right)^K \right). \quad (82)$$

The server can use either the proposed scheme or the conventional scheme, whichever uses the minimal rate. Thus combining (79) and (82), Algorithm 2 achieves a rate of:

$$R_s^{D}(M) = K \left( 1 - \frac{M-1}{N-1} \right) \min \left\{ \frac{N-1}{K(M-1)} \cdot \left( 1 - \left( 1- \frac{M-1}{N-1} \right)^K \right), 1, \frac{N}{K} \right\}. \quad (83)$$

Using the fact that

$$\left( 1 - \frac{M-1}{N-1} \right)^K \geq 1 - \frac{K(M-1)}{N-1},$$

the expression in (83) can be simplified to give the expression in Theorem 5

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

**Proof of Secure Achievability**

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

$$I \left( X_{(d_1, \ldots, d_K)}; W_1, \ldots, W_N \right) = 0 \quad (85)$$

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 (71), consists of $\binom{K}{s}$ transmissions for each $s = K, K -$
1, \ldots, 1. We have:

\[
I \left( X_{(d_1, \ldots, d_K)}; W_1, \ldots, W_N \right) = H \left( X_{(d_1, \ldots, d_K)} \right) - H \left( X_{(d_1, \ldots, d_K)} | W_1, \ldots, W_N \right)
\]

\[
= H \left( X_{(d_1, \ldots, d_K)} \right) - H \left( \left\{ X^s_{(d_1, \ldots, d_K)} \right\}_{s=1}^K | W_1, \ldots, W_N \right)
\]

\[
= H \left( X_{(d_1, \ldots, d_K)} \right) - H \left( \left\{ \mathcal{K}_S \oplus_k \mathcal{S} \cdot W_{d_k, S \setminus \{k\}} : |S| = s \right\}_{s=1}^K | W_1, \ldots, W_N \right)
\]

\[
= H \left( X_{(d_1, \ldots, d_K)} \right) - H \left( \left\{ \mathcal{K}_S : |S| = s \right\}_{s=1}^K \left| W_1, \ldots, W_N \right. \right)
\]

\[
= H \left( X_{(d_1, \ldots, d_K)} \right) - H \left( \left\{ \mathcal{K}_S : |S| = s \right\}_{s=1}^K \right), \tag{86}
\]

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 \( H(A, B) \leq H(A) + H(B) \), we have:

\[
H \left( X_{(d_1, \ldots, d_K)} \right) = H \left( \left\{ X^s_{(d_1, \ldots, d_K)} \right\}_{s=1}^K \right) \leq \sum_{s=1}^K H \left( X^s_{(d_1, \ldots, d_K)} \right)
\]

\[
\leq \sum_{s=1}^K \sum_{i=1}^K H \left( \mathcal{K}_S, \oplus_{k \in S_i} W_{d_k, S_i \setminus \{k\}} : |S| = s \right) \leq \sum_{s=1}^K \sum_{i=1}^K \log_2 \left( F q^{s-1} (1 - q)^{K-s+1} \right)
\]

\[
= \sum_{s=1}^K \left( \begin{array}{c} K \vspace{1mm} \\ s \end{array} \right) \log_2 \left( F q^{s-1} (1 - q)^{K-s+1} \right). \tag{87}
\]

On the other hand, we have:

\[
H \left( \left\{ \mathcal{K}_S : |S| = s \right\}_{s=1}^K \right) = \sum_{s=1}^K H \left( \left\{ \mathcal{K}_S : |S| = s \right\} \right) = \sum_{s=1}^K \sum_{i=1}^K H \left( \mathcal{K}_{S_i} : |S_i| = s \right)
\]

\[
= \sum_{s=1}^K \sum_{i=1}^K \log_2 \left( F q^{s-1} (1 - q)^{K-s+1} \right) = \sum_{s=1}^K \left( \begin{array}{c} K \vspace{1mm} \\ s \end{array} \right) \log_2 \left( F q^{s-1} (1 - q)^{K-s+1} \right), \tag{88}
\]

where the equality in \(\text{(88)}\) follows from the fact that the keys are orthogonal to each other and they are uniformly distributed as in \(\text{(70)}\). Substituting \(\text{(87)}\) and \(\text{(88)}\) into \(\text{(86)}\), we have:

\[
I \left( X_{(d_1, \ldots, d_K)}; W_1, \ldots, W_N \right) \leq 0 \tag{89}
\]

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

\[
I \left( X_{(d_1, \ldots, d_K)}; W_1, \ldots, W_N \right) = 0 \tag{90}
\]

which proves that the rate \( R_s^D(M) \) is securely achievable. This completes the proof of Theorem \[5\] \(\square\)

APPENDIX F
PROOF OF THEOREM \[6\]

From Theorem \[5\], we have:

\[
R_s^D(M) = K \left( 1 - \frac{M - 1}{N - 1} \right) \min \left\{ \frac{N - 1}{K(M - 1)} \cdot \left( 1 - \left( 1 - \frac{M - 1}{N - 1} \right)^K \right)^{-1} \right\} \cdot \frac{N}{K} \tag{91}
\]

\[^3\text{Equality holds only if A and B are orthogonal (independent) to each other \cite{12}.}\]
Using the fact that
\[
\left(1 - \frac{M - 1}{N - 1}\right)^K \geq 1 - \frac{K(M - 1)}{N - 1}
\]
and
\[
\left(1 - \frac{M - 1}{N - 1}\right)^K \geq 0
\]
the rate can be upper bounded as:
\[
R^D_s(M) \leq \min\left\{\frac{N - 1}{M - 1} - 1, K\left(1 - \frac{M - 1}{N - 1}\right), N\left(1 - \frac{M - 1}{N - 1}\right)\right\}, \quad (92)
\]
for all \(M \in \frac{N - 1}{N} \cdot (0, N] + 1\). We consider two cases: \(\min\{N, K\} \leq 17\) and \(\min\{N, K\} \geq 18\). First considering \(\min\{N, K\} \leq 17\), from (92), we have:
\[
R^D_s(M) \leq \min\{N, K\} \left(1 - \frac{M - 1}{N - 1}\right). \quad (93)
\]
Also, setting \(s = 1\) in Theorem 3 gives:
\[
R^*_s(M) \geq \left(1 - \frac{M - 1}{N - 1}\right). \quad (94)
\]
Thus we have:
\[
\frac{R^D_s(M)}{R^*_s(M)} \leq \min\{N, K\} \leq 17. \quad (95)
\]
For \(\min\{N, K\} \geq 18\), we consider 3 distinct regimes:

**Regime 1:** \(0 \leq M - 1 \leq 1.2 \max \left(1, \frac{N - 1}{K}\right)\)

**Regime 2:** \(1.2 \max \left(1, \frac{N - 1}{K}\right) < M - 1 \leq \frac{(N - 1)}{17}\)

**Regime 3:** \(\frac{(N - 1)}{17} < M - 1 \leq N - 1\)

We consider each of the three regimes separately.

**Regime 1:** \(0 \leq M - 1 \leq 1.2 \max \left(1, \frac{N - 1}{K}\right)\)

By (92), we have:
\[
R^D_s(M) \leq R^D_s(1) \leq \min\{N, K\}. \quad (96)
\]
By Theorem 3 and using the fact that \([N/s] \geq N/s - 1\), we have:
\[
R^*_s(M) \geq s - \frac{s^2(M - 1)}{N - 2s}. \quad (97)
\]
Setting \(s = \lfloor 0.1586 \min\{N, K\} \rfloor\) we get, for \(M - 1 \leq 1.2 \max \left(1, \frac{N - 1}{K}\right)\):
\[
R^*_s(M) \geq R^*_s\left(1.2 \max \left(1, \frac{N - 1}{K}\right) + 1\right) \geq 0.1586 \min\{N, K\} - 1 - \frac{(0.1586 \min\{N, K\})^2 \cdot 1.2 \max \left(1, \frac{N - 1}{K}\right)}{N - 2 \cdot 0.1586 \min\{N, K\}} \geq \min\{N, K\} \left\{0.1586 - \frac{1}{\min\{N, K\}} - \frac{(0.1586)^2 \cdot 1.2}{1 - 2 \cdot (0.1586) \min\{1, K/N\}}\right\} \geq \min\{N, K\} \left\{0.1586 - \frac{1}{18} - \frac{1 - 2 \cdot (0.1586)^2}{1 - 2 \cdot 0.1586}\right\} \geq \frac{1}{17} \min\{N, K\}. \quad (98)
\]
Combining (96) and (98), we get:
\[
\frac{R^D_s(M)}{R^*_s(M)} \leq 17.
\] (99)

**Regime 2:** \(1.2 \max \left(1, \frac{N-1}{M} \right) < M - 1 \leq \frac{(N-1)}{M}\)

Using (92), we have:
\[
R^D_s(M) \leq \frac{N - 1}{M - 1} - 1 \leq \frac{N - 1}{M - 1}.
\] (100)

Now setting \(s = \lfloor 0.1460 \frac{N-1}{M-1} \rfloor\) in Theorem 3 we have:
\[
R^*_s(M) \geq 0.1460 \frac{N - 1}{M - 1} - 1 - \frac{0.1460^2 \cdot \frac{N - 1}{M - 1}^2}{N - 2} - \frac{0.1460 \cdot \frac{N - 1}{M - 1}}{M - 1}
\]
\[
\geq \frac{N - 1}{M - 1} \left\{ 0.0.1460 - \frac{1}{17} - \frac{0.1460^2}{1.2} \right\}
\]
\[
\geq \frac{1}{17} \left( \frac{N - 1}{M - 1} \right).
\] (101)

Combining (100) and (101), we get:
\[
\frac{R^D_s(M)}{R^*_s(M)} \leq 17.
\] (102)

**Regime 3:** \(\frac{(N-1)}{17} < M - 1 \leq N - 1\)

From (92), we have:
\[
R^D_s(M) \leq \frac{N - 1}{M - 1} - 1.
\] (103)

Setting \(s = 1\) in Theorem 3 we have again:
\[
R^*_s(M) \geq \left( 1 - \frac{M - 1}{N - 1} \right).
\] (104)

Thus combining (103) and (104), we get:
\[
\frac{R^D_s(M)}{R^*_s(M)} \leq \frac{\frac{N - 1}{M - 1} - 1}{1 - \frac{M - 1}{N - 1}}
\]
\[
= \frac{N - 1}{M - 1} \leq 17.
\] (105)

Thus we have proved that for any \(N,K \in \mathbb{N}\) and all \(1 \leq M \leq N\), there is a constant multiplicative gap of 17 between the achievable secure decentralized rate and the information theoretic optimal for any secure scheme. This concludes the proof of Theorem 6.

\[\square\]

**REFERENCES**

[1] Lawrence W. Dowdy and Derrell V. Foster, “Comparative models of the file assignment problem,” *ACM Comput. Surv.*, vol. 14, no. 2, pp. 287–313, June 1982.

[2] Kevin C. Almeroth and Mostafa H. Ammar, “The use of multicast delivery to provide a scalable and interactive video-on-demand service,” *Selected Areas in Communications, IEEE Journal on*, vol. 14, no. 6, pp. 1110–1122, 1996.

[3] Madhukar R Korupolu, C.Greg Plaxton, and Rajmohan Rajaraman, “Placement algorithms for hierarchical cooperative caching,” *Journal of Algorithms*, vol. 38, no. 1, pp. 260 – 302, 2001.

[4] Ivan Baev, Rajmohan Rajaraman, and Chaitanya Swamy, “Approximation algorithms for data placement problems,” *SIAM J. Comput.*, vol. 38, no. 4, pp. 1411–1429, Aug. 2008.

[5] Mohammad Ali Maddah-Ali and Urs Niesen, “Fundamental limits of caching,” *arXiv: 1209.5807*, Sep 2012.

[6] U. Niesen and M. A. Maddah-Ali, “Coded Caching with Nonuniform Demands,” *arXiv : 1308.0178*, Aug 2013.
[7] Mohammad Ali Maddah-Ali and Urs Niesen, “Decentralized caching attains order-optimal memory-rate tradeoff,” arXiv:1301.5848, Jan 2013.

[8] L. Breslau, Pei Cao, Li Fan, G. Phillips, and S. Shenker, “Web caching and zipf-like distributions: evidence and implications,” in IEEE INFOCOM, 1999, vol. 1, pp. 126–134 vol.1.

[9] Negin Golrezaei, Alexandros G. Dimakis, and Andreas F. Molisch, “Wireless device-to-device communications with distributed caching,” arXiv:1205.7044, May 2012.

[10] Mingyue Ji, Giuseppe Caire, and Andreas F. Molisch, “Optimal throughput-outage trade-off in wireless one-hop caching networks,” arXiv:1302.2168, Feb 2013.

[11] C. E. Shannon, “Communication theory of secrecy systems,” Bell System Technical Journal, vol. 28, no. 4, pp. 656–715, 1949.

[12] Thomas M. Cover and Joy A. Thomas, Elements of information theory, Wiley-Interscience, New York, NY, USA, 1991.