2016 IEEE Conference on Communications and Network Security (CNS): The Workshop on Physical-Layer Methods for Wireless Security

Coded Caching for Resolvable Networks with Security Requirements

Ahmed A. Zewail and Aylin Yener

Wireless Communications and Networking Laboratory (WCAN)
Electrical Engineering Department
The Pennsylvania State University, University Park, PA 16802.
zewail@psu.edu yener@engr.psu.edu

***Abstract*—We consider networks where a set of end users, equipped with cache memories, are connected to a single server via a set of relay nodes. We further consider a special class of such networks that satisfy the so called resolvability property and study centralized coded caching scenarios under three security requirements. Under the first scenario, i.e., secure delivery, the wiretapper should not gain any information about the database files from the transmitted signals over the network links. In the second scenario, i.e., secure coded caching, we consider the case where users should not be able to obtain any information about files that they did not request. In the third scenario, the system requires both secure delivery and secure coded caching. We provide achievable schemes by jointly optimizing the cache placement and delivery phases, utilizing secret sharing and one-time pads. We provide numerical results to compare performance under these different requirements.**

I. Introduction

Caching is an effective technique to avoid congestion in wireless communication networks during peak traffic times. By preallocating functions of data contents in cache memories of the end users during the low traffic hours, known as *the cache placement phase*, the needed transmission rates during peak traffic, when the users actually request the files, can be significantly reduced. Caching alleviates the need to download the contents that have already been placed in the user's cache, and allows the system to create multicast opportunities, during *the delivery phase*. References [1]–[3] have characterized the fundamental limits of caching systems which represent the trade-off between the cache size and the transmission rate, to satisfy the users' requests under different requirements. These references focus on one-hop scenarios where a single server is connected to the end users via a shared link. Recently, references [4]–[7] have studied cache-aided systems considering more elaborate network structures. In particular, references [5] and [6] investigated a single server symmetric layered network, known as a *combination network*, where the end users have caching capabilities. In such a network, the server is connected to a set of h relay nodes, which communicate to $\binom{h}{r}$ users. Each user is connected to r relay nodes, and is equipped with a cache memory. In these references, users randomly cache a fraction of bits from each file, i.e., employ decentralized coded caching [8]. More recently, reference [7] has considered a class of networks that satisfies the resolvability property. This class includes the combination networks [5], [6], whenever r divides h. In contrast with [5] and [6], reference [7] considered centralized coded caching [1] and proposed a delivery strategy that outperforms those in [5] and [6].

In this work, we investigate such a network topology and quantify the performance of the cache-aided network under different security requirements. In particular, we consider the model similar to the one in [7] under three different scenarios. First, we consider a network where the files should be kept secret from any external eavesdropper that overhears the delivery phase. This requirement is known as *secure delivery* [2]. We jointly optimize the cache placement and delivery phases, utilizing random keys, to derive an upper bound on the required transmission rate. Second, we consider a scenario where each user should only be able to decode its requested file and should not be able gain any information about the contents of the remaining files [3]. We refer to this requirement as *secure coded caching*. In the proposed achievability scheme, we utilize secret sharing techniques from [9]. In particular, for a file W with size F bits, an (m, n) secret sharing scheme generates n shares, $S_1, S_2, ..S_n$, such that accessing any m shares does not reveal any information about W, i.e.,

$$I(W; \mathcal{S}) = 0, \quad \forall \mathcal{S} \subseteq \{S_1, S_2, ..S_n\}, |\mathcal{S}| \leq m. \qquad (1)$$

Furthermore, W can be losslessly reconstructed from the n shares, i.e., $H(W|S_1, S_2, .., S_n) = 0$. For large enough F, an (m, n) secret sharing scheme exists with shares of size equal to $\frac{F}{n-m}$ bits [9]. Third, we consider a scenario where both secure delivery and secure coded caching are required, simultaneously. Here, the achievability relies on both secret sharing schemes and one-time pads.

Our study demonstrates the impact of the network topology and structure on system performance under security requirements. In addition to the benefit from lowering the subpacketization level as illustrated in [7], we show that satisfying the secure coded caching requirement does not require additional memory at the end users nor encryption keys, unlike the case in [3]. Moreover, we observe that the cost due the secure delivery requirement is almost negligible, similar to the case in [2].

It is worth mentioning that the considered setup can model a layered wireless network where end users are served by small cell base stations, such that each user is simultaneously

978-1-5090-3065-1/16/$31.00 ©2016 IEEE

connected to r base stations via orthogonal channels [5]. The scenarios with secure delivery requirement can model systems where the base stations are untrusted, i.e., honest-but-curious, as in [10], [11].

The remainder of the paper is organized as follows. In Section II, we describe the system model. In Sections III, IV and V, we detail the achievability techniques for the different security requirements. In Section VI, we discuss the learned insights from our work. Section VII concludes the paper.

II. System Model

We consider a two-hop network, where the server, S, is connected to K end users via a set of h relay nodes. More specifically, each end user is connected to a distinct set of $r < h$ relay nodes. Let $\mathcal{R} = \{\Gamma_1, .., \Gamma_h\}$ denote the set of relay nodes. Each end user is denoted by $U_{\mathcal{V}}$, where $\mathcal{V}$ is the set of relays' indices which are connected to that end user, i.e., $|\mathcal{V}| = r$. In addition, we define $\mathcal{U}$ to represent the set of all end users in the network, and $\mathcal{V}_{\mathcal{U}}$ to be the set of all subsets that specify the end users, i.e., if $U_{\mathcal{V}} \in \mathcal{U}$, then $\mathcal{V} \in \mathcal{V}_{\mathcal{U}}$. We consider resolvable networks [7], whose definition we recall below for the sake of completeness.

Definition 1. *The set $\mathcal{V}_{\mathcal{U}}$ is said to be resolvable if there exists a partition of $\mathcal{V}_{\mathcal{U}}$ into subsets $\mathcal{P}_1$, $\mathcal{P}_2$,.., $\mathcal{P}_{\hat{K}}$, known as the parallel classes of $\mathcal{V}_{\mathcal{U}}$, such that for any $i \in \{1, .., \hat{K}\}$,*

1) if $\mathcal{V} \in \mathcal{P}_i$ and $\bar{\mathcal{V}} \in \mathcal{P}_i$, then $\mathcal{V} \cap \bar{\mathcal{V}} = \phi$, and

2) $\cup_{\mathcal{V}:\mathcal{V}\in\mathcal{P}_i} \mathcal{V} = \{1, .., h\}$. ■

We denote the set of end users connected to Γ_i by $\mathcal{N}(\Gamma_i)$. It can be verified that for a resolvable network $|\mathcal{N}(\Gamma_i)| = \frac{Kr}{h} = \hat{K}$. Also, it is clear that each user belongs to exactly one parallel class $\mathcal{P}_i$. Let $\Delta(\mathcal{V})$ denote the parallel class that user $U_{\mathcal{V}}$ belongs to. We define the groups $\mathcal{G}_i$, where $\mathcal{G}_i = \{U_{\mathcal{V}} : \mathcal{V} \in \mathcal{P}_i\}$ and $i = 1, .., \hat{K}$. In addition, we define $\mathcal{V}[i]$ to represent the i-th element in $\mathcal{V}$, assuming that $\mathcal{V}$'s elements are ordered, e.g., if $\mathcal{V} = \{3, 7\}$, then $\mathcal{V}[1] = 3$ and $\mathcal{V}[2] = 7$. Let $Inv(\mathcal{V}[i])$ be the inverse mapping of $\mathcal{V}[i]$, i.e., $Inv(\mathcal{V}[i]) = j$ if $\mathcal{V}[j] = i$. We illustrate the aforementioned property and notation by the following example.

Example 1. Consider the network, depicted in Fig. 1, where $h = 4$ and $K = 6$. Each end user is connected to two relay nodes, i.e., $r = 2$. Note that the first end user on the left is denoted by U_{12} as it is connected to Γ_1 and Γ_2. The parallel classes that represent the end users are,

$$\mathcal{P}_1 = \{\{1,2\},\{3,4\}\}\,,\ \mathcal{P}_2 = \{\{1,3\},\{2,4\}\}\,, \\ \text{and } \mathcal{P}_3 = \{\{1,4\},\{2,3\}\}\,.$$

The end users are partitioned into $\hat{K} = 3$ groups, given by

$$\mathcal{G}_1 = \{U_{12}, U_{34}\}\,, \mathcal{G}_2 = \{U_{13}, U_{24}\}\,, \text{and } \mathcal{G}_3 = \{U_{14}, U_{23}\}\,.$$

■

The server S has a database of N files, $W_1, .., W_N$, each with size F bits. Similar to references [1]–[7], all network links are assumed to be noiseless. Each user is equipped with a cache memory with size MF bits, i.e., M is the normalized cache memory size. The system operates over two phases.

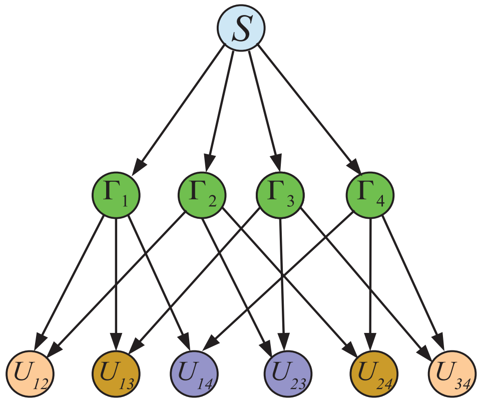

Fig. 1: A resolvable network with $K = 6$, $h = 4$ and $r = 2$. End users are partitioned into 3 groups, each represented by a different color.

A. Cache placement phase

In this phase, the server allocates functions of its database in the end users' cache. These allocations are designed without the knowledge of its requested files in the near future.

Definition 2. *(Cache Placement): The content of the cache memory at user $U_{\mathcal{V}}$ is given by*

$$Z_{\mathcal{V}} = \phi_{\mathcal{V}}(W_1, W_2, .., W_N), \tag{2}$$

where $\phi_{\mathcal{V}} : [2^F]^N \to [2^F]^M$, i.e., $H(Z_{\mathcal{V}}) \le MF$. ■

B. Delivery phase

During peak traffic, each user requests a randomly selected file [1]. We define $d_{\mathcal{V}}$ to denote the index of the requested file by $U_{\mathcal{V}}$, i.e., $d_{\mathcal{V}} \in \{1, 2, .., N\}$, and $\boldsymbol{d}$ to represent the demand vector of all network users at any request instance. The server responds to the users' requests by transmitting signals to each of the relay nodes. Then, each relay forwards its received signal to the set of intended end users. From the r received signals and $Z_{\mathcal{V}}$, user $U_{\mathcal{V}}$ reconstructs its requested file $W_{d_{\mathcal{V}}}$.

Definition 3. *(Coded Delivery): The mapping from the database, and the demand vector $\mathbf{d}$ into the transmitted signal by the server to Γ_k is represented by the encoding function*

$$X_{k,\boldsymbol{d}} = \psi_k(W_1, .., W_N, \boldsymbol{d}), \qquad k = 1, 2, .., h, \tag{3}$$

where $\psi_k : [2^F]^N \times \{1, ..N\}^K \to [2^F]^{R_1}$, and R_1 is the rate, normalized by the file size, F, of the transmitted signal from the server to each relay node. Also, the transmitted signal from Γ_k to user $U_{\mathcal{V}} \in \mathcal{N}(\Gamma_k)$, is defined by the encoding function

$$Y_{k,\boldsymbol{d},\mathcal{V}} = \varphi_{\mathcal{V}}(X_{k,\boldsymbol{d}}, \boldsymbol{d}), \tag{4}$$

where $\varphi_{\mathcal{V}} : [2^F]^{R_1} \times \{1, ..N\}^K \to [2^F]^{R_2}$, and R_2 is the normalized rate of the transmitted signal from the relay node *to a connected end user. In addition, user $U_{\mathcal{V}}$ has a decoding function to recover its requested file*

$$\hat{W}_{d_{\mathcal{V}}} = \mu_{\mathcal{V}}(Z_{\mathcal{V}}, \boldsymbol{d}, \{Y_{k,\boldsymbol{d},\mathcal{V}} : k \in \mathcal{V}\}), \tag{5}$$

where $\mu_{\mathcal{V}} : [2^F]^M \times \{1,..N\}^K \times [2^F]^{R_2} \to [2^F]$. ■

Each of the end users must be able to recover its requested file reliably, i.e., for any $\epsilon > 0$,

$$\max_{\boldsymbol{d},\mathcal{V}} P(\hat{W}_{d_{\mathcal{V}}} \neq W_{d_{\mathcal{V}}}) < \epsilon. \tag{6}$$

III. Coded caching with secure delivery

In our first scenario, we study the network described in Section II under the secure delivery requirement. In particular, we require that any eavesdropper that observes the transmitted signals, during the delivery phase, should not be able to gain any information about the files, i.e., for any $\delta > 0$

$$I(\mathcal{X}, \mathcal{Y}; W_1, ..W_N) < \delta, \tag{7}$$

where $\mathcal{X}, \mathcal{Y}$ are the sets of transmitted signals by the server and the relay nodes, respectively.

A. Cache placement phase

For $M = 1 + t\frac{N-1}{\hat{K}}$, and $t \in \{0, 1, .., \hat{K}\}$, each file in the database is divided into $r\binom{\hat{K}}{t}$ disjoint parts each of which is denoted by $W^j_{n,\mathcal{T}}$, where n is the file index i.e., $n \in \{1, .., N\}$, $j = 1, .., r$, and $\mathcal{T} \subseteq \{1, .., \hat{K}\}, |\mathcal{T}| = t$. The server allocates the subfiles $W^j_{n,\mathcal{T}}$, $\forall j, n$ in the cache memory of user $U_{\mathcal{V}} \in \mathcal{G}_i$ if $i \in \mathcal{T}$. Furthermore, the server generates $h\binom{\hat{K}}{t+1}$ independent keys. Each key is denoted by $K^u_{\mathcal{T}_K}$, where $u = 1, .., h$, and $\mathcal{T}_K \subseteq \{1, .., \hat{K}\}, |\mathcal{T}_K| = t + 1$. User $U_{\mathcal{V}} \in \mathcal{G}_i$ stores the keys $K^u_{\mathcal{T}_K}$, $\forall u \in \mathcal{V}$, whenever $i \in \mathcal{T}_K$, i.e.,

$$Z_{\mathcal{V}} = \{W^j_{n,\mathcal{T}}, K^u_{\mathcal{T}_K} : i \in \mathcal{T}, \forall n, j, \text{ and } i \in \mathcal{T}_K, \forall u \in \mathcal{V}\}. \tag{8}$$

It can be verified that this allocation satisfies the memory capacity constraint, and yields

$$t = \frac{\hat{K}(M-1)}{N-1} = \frac{Kr(M-1)}{h(N-1)}. \tag{9}$$

B. Coded Delivery phase

At the beginning of the delivery phase, the demand vector $\boldsymbol{d}$ is announced in the network as public information. For each relay Γ_i, at each transmission instance, we consider $\mathcal{S} \subseteq \mathcal{N}(\Gamma_i)$, where $|\mathcal{S}| = t+1$. For each $\mathcal{S}$, the server transmits to the relay node Γ_i, the following signal

$$X^{\mathcal{S}}_{i,\boldsymbol{d}} = K^i_{\mathcal{S}} \oplus_{\{\mathcal{V}:\Delta(\mathcal{V})\in\mathcal{S}\}} W^{Inv(\mathcal{V}[i])}_{d_{\mathcal{V}},\mathcal{S}\setminus\{\Delta(\mathcal{V})\}}. \tag{10}$$

In total, the server transmits to Γ_i, the following signal

$$X_{i,\boldsymbol{d}} = \cup_{\mathcal{S}\subseteq\mathcal{N}(\Gamma_i):|\mathcal{S}|=t+1}\{X^{\mathcal{S}}_{i,\boldsymbol{d}}\}. \tag{11}$$

Then, Γ_i forwards the signal $X^{\mathcal{S}}_{i,\boldsymbol{d}}$ to the users in the set $\mathcal{S}$. The user $U_{\mathcal{V}} \in \mathcal{S}$ can recover the following set of subfiles from the signals received from Γ_i, utilizing its cache's contents

$$\{W^{Inv(\mathcal{V}[i])}_{d_{\mathcal{V}},\mathcal{T}} : \mathcal{T} \subset \{1, .., \hat{K}\} \setminus \{\Delta(\mathcal{V})\}, |\mathcal{T}| = t\}.$$

Since, each user receives signals from r relays, it obtains

$$\cup_{i\in\mathcal{V}}\{W^{Inv(\mathcal{V}[i])}_{d_{\mathcal{V}},\mathcal{T}} : \mathcal{T} \subset \{1, .., \hat{K}\} \setminus \{\Delta(\mathcal{V})\}, |\mathcal{T}| = t\}.$$

Utilizing the contents of $Z_{\mathcal{V}}$, $U_{\mathcal{V}}$ is able to reconstruct $W_{d_{\mathcal{V}}}$.

Now, we calculate the transmission rates resulted from the aforementioned scheme. Under secure delivery, we refer to R_1 and R_2 as R^s_1 and R^s_2, respectively. Observe that each relay is responsible for $\binom{\hat{K}}{t+1}$ transmissions, each of length $\frac{F}{r\binom{\hat{K}}{t}}$, thus the transmission rate in bits from the server to each relay is

$$R^s_1 F = \frac{\binom{\hat{K}}{t+1}}{r\binom{\hat{K}}{t}} F = \frac{\hat{K} - t}{r(t+1)} F = \frac{\hat{K}\left(1 - \frac{M-1}{N-1}\right)}{r\left(\hat{K}\frac{M-1}{N-1} + 1\right)} F. \tag{12}$$

In addition, each relay forwards $\binom{\hat{K}-1}{t}$ from its received signals to each of its connected end users, thus

$$R^s_2 F = \frac{\binom{\hat{K}-1}{t}}{r\binom{\hat{K}}{t}} F = \frac{\hat{K} - t}{r\hat{K}} F = \frac{\left(1 - \frac{M-1}{N-1}\right)}{r} F. \tag{13}$$

Therefore, we can obtain the following upper bound on the normalized transmission rates.

Theorem 1. *The normalized transmission rates with secure delivery, for $M = 1 + \frac{th}{Kr}(N-1)$, and $t \in \{0, 1, .., \frac{Kr}{h}\}$, are upper bounded by*

$$R^s_1 \leq \frac{K\left(1 - \frac{M-1}{N-1}\right)}{h\left(\frac{Kr(M-1)}{h(N-1)} + 1\right)}, \quad R^s_2 \leq \frac{1}{r}\left(1 - \frac{M-1}{N-1}\right). \tag{14}$$

The convex envelope of these points is achievable. □

We note that, if M is not in the form of $1 + \frac{th(N-1)}{Kr}$, we apply memory sharing as in [1] for achievability.

Remark 1. *In total, the server sends $h\binom{\hat{K}}{t+1}$ signals, each of which is encrypted using a one-time pad that has length equal to the length of each subfile. Therefore, observing any of the transmitted signals without the knowledge of the encryption key will not reveal any information about the database files. In other words, the above achievability scheme ensures that condition (7) is satisfied.* ■

IV. Secure coded caching

In this section, we investigate the network under the secure coded caching requirement. In particular, an end user should only be able to recover its requested file, and should *not* be able to obtain any information about the remaining files, i.e., for $\delta > 0$

$$\max_{\boldsymbol{d},\mathcal{V}} I(\boldsymbol{W}_{-d_{\mathcal{V}}}; \{Y_{k,\boldsymbol{d},\mathcal{V}} : k \in \mathcal{V}\}, Z_{\mathcal{V}}) < \delta, \tag{15}$$

where $\boldsymbol{W}_{-d_{\mathcal{V}}} = \{W_1, .., W_N\} \setminus \{W_{d_{\mathcal{V}}}\}$, i.e., the set of all files except the one requested by user $U_{\mathcal{V}}$.

A. Cache placement phase

For $M = \frac{tN}{\hat{K}-t}$, and $t \in \{0, 1, .., \hat{K} - 1\}$, each file is encoded using the $\left(r\binom{\hat{K}}{t}, r\binom{\hat{K}-1}{t-1}\right)$ secret sharing scheme from [9]. The resulting shares are denoted by $S^{j}_{n,\mathcal{T}}$, where n is the file index i.e., $n \in \{1,..,N\}$, $j = 1,..,r$, and $\mathcal{T} \subseteq \{1,..,\hat{K}\}, |\mathcal{T}| = t$. Each share has size

$$F_s = \frac{F}{r\binom{\hat{K}}{t} - r\binom{\hat{K}-1}{t-1}} = \frac{tF}{r(\hat{K}-t)\binom{\hat{K}-1}{t-1}} \text{ bits.} \tag{16}$$

The server allocates the shares $S^{j}_{n,\mathcal{T}}$, $\forall j, n$ in the cache of user $U_{\mathcal{V}} \in \mathcal{G}_i$ whenever $i \in \mathcal{T}$. Such allocation agrees with the memory capacity constraint, thus we have

$$t = \frac{\hat{K}M}{N+M} = \frac{KrM}{h(N+M)}. \tag{17}$$

B. Coded Delivery phase

At the beginning of the delivery phase, each user requests a file from the server. First, we focus on the transmissions from the server to Γ_i. At each transmission instance, we consider $\mathcal{S} \subseteq \mathcal{N}(\Gamma_i)$, where $|\mathcal{S}| = t{+}1$. For each $\mathcal{S}$, the server transmits the following signal to Γ_i

$$X^{\mathcal{S}}_{i,\boldsymbol{d}} = \oplus_{\{\mathcal{V}:\Delta(\mathcal{V})\in\mathcal{S}\}} S^{Inv(\mathcal{V}[i])}_{d_{\mathcal{V}},\mathcal{S}\setminus\{\Delta(\mathcal{V})\}}. \tag{18}$$

Thus, $X_{i,\boldsymbol{d}} = \cup_{\mathcal{S}\subseteq\mathcal{N}(\Gamma_i):|\mathcal{S}|=t+1}\{X^{\mathcal{S}}_{i,\boldsymbol{d}}\}$. Γ_i forwards $X^{\mathcal{S}}_{i,\boldsymbol{d}}$ to the users in the set $\mathcal{S}$. Since, each user receives signals from r relays, it can obtain the shares

$$\cup_{i\in\mathcal{V}}\{S^{Inv(\mathcal{V}[i])}_{d_{\mathcal{V}},\mathcal{T}} : \mathcal{T} \subset \{1,..,\hat{K}\} \setminus \{\Delta(\mathcal{V})\}, |\mathcal{T}| = t\}. \tag{19}$$

Thus, user $U_{\mathcal{V}}$ recovers its requested file from its $r\binom{\hat{K}}{t}$ shares.

Next, we define the transmission rates of this scheme. To distinguish between different requirements, under secure coded caching requirement, we refer to R_1 and R_2 as R^c_1 and R^c_2, respectively. Since, each relay is responsible for $\binom{\hat{K}}{t+1}$ transmissions, each of length F_s, the transmission rate in bits from the server to each relay is

$$R^c_1 F = \frac{t\binom{\hat{K}}{t+1}F}{r(\hat{K}-t)\binom{\hat{K}-1}{t-1}} = \frac{\hat{K}F}{r(t+1)} = \frac{\hat{K}(N+M)F}{r\left((\hat{K}+1)M+N\right)}. \tag{20}$$

On the other hand, each relay forwards $\binom{\hat{K}-1}{t}$ from its received signals to each of its connected end users, therefore

$$R^c_2 F = \frac{t\binom{\hat{K}-1}{t}}{r(\hat{K}-t)\binom{\hat{K}-1}{t-1}} F = \frac{1}{r}F. \tag{21}$$

Consequently, we have the following theorem.

Theorem 2. *The normalized rates with secure coded caching, for $M{=}\frac{tNh}{Kr-th}$ and $t{\in}\{0,1,..,\frac{Kr}{h}{-}1\}$, are upper bounded by*

$$R^c_1 \leq \frac{K(N+M)}{M(Kr+h)+Nh}, \quad R^c_2 \leq \frac{1}{r}. \tag{22}$$

The convex envelope of these points is achievable. □

Using memory sharing techniques, explained in [1], we can achieve the convex envelope of the points given by the values $M = \frac{tNh}{Kr-th}$, and $t \in \{0,1,..,\frac{Kr}{h}-1\}$. This concludes the proof of Theorem 2.

Remark 2. *Secret sharing encoding guarantees that no user is able to reconstruct any file from its cache contents only. In addition, the only new information in the received signals by any user, as expressed in (18), is the shares of its requested file, i.e., the secure coded caching requirement is satisfied.* ■

V. Secure coded caching with secure delivery

In this section, we investigate the network under the requirements studied in Sections III and IV. In particular, any end user should not obtain any information about the database files that he did not request, while any external eavesdropper should not obtain any information about the content of the database files from overhearing the delivery phase.

A. Cache placement phase

For $M = \frac{tN}{\hat{K}-t} + 1$, and $t \in \{0,1,..,\hat{K}-1\}$, each file encoded using secret sharing scheme with the same parameters as described in subsection IV-A and the resultant shares are allocated as described before in the end users' memories. Furthermore, the server generates $h\binom{\hat{K}}{t+1}$ independent keys, each of length F_s bits and denoted by $K^{u}_{\mathcal{T}_K}$, where $u{=}1,..,h$, and $\mathcal{T}_K \subseteq \{1,..,\hat{K}\}, |\mathcal{T}_K| = t{+}1$. User $U_{\mathcal{V}}$ stores the keys $K^{u}_{\mathcal{T}_K}$, $\forall u \in \mathcal{V}$, whenever it belongs to $\mathcal{G}_i$ and $i \in \mathcal{T}$. This allocation satisfies the memory constraint and gives

$$t = \frac{\hat{K}(M-1)}{N+M-1} = \frac{Kr(M-1)}{h(N+M-1)}. \tag{23}$$

B. Coded Delivery phase

Once the demand vector $\boldsymbol{d}$ is announced in the network, the delivery phase starts. For Γ_i, at each transmission instance, the system serves a set $\mathcal{S} \subseteq \mathcal{N}(\Gamma_i)$, where $|\mathcal{S}| = t+1$. For each $\mathcal{S}$, the server transmits to Γ_i, the following signal

$$X^{\mathcal{S}}_{i,\boldsymbol{d}} = K^{i}_{\mathcal{S}} \oplus_{\{\mathcal{V}:\Delta(\mathcal{V})\in\mathcal{S}\}} S^{Inv(\mathcal{V}[i])}_{d_{\mathcal{V}},\mathcal{S}\setminus\{\Delta(\mathcal{V})\}}. \tag{24}$$

Utilizing its cache memory and the received signals from the r connected relay nodes, it can be seen that each user is able to recover its requested file from its $r\binom{\hat{K}}{t}$ shares. By calculating the transmission rates, we can obtain the following upper bound on the normalized transmission rates.

Theorem 3. *Under secure delivery and secure coded caching requirements, for $M{=}\frac{tNh}{Kr-th}{+}1$ and $t \in \{0,1,..,\frac{Kr}{h}{-}1\}$, the transmission rates are upper bounded by*

$$R^{sc}_1 \leq \frac{K(N+M-1)}{(M-1)(Kr+h)+Nh}, \quad R^{sc}_2 \leq \frac{1}{r}. \tag{25}$$

The convex envelope of these points is achievable. □

It is worth noting that, in order to satisfy the secure delivery requirement, each user stores $r\binom{\hat{K}-1}{t}$ keys that consume F bits from its memory. Therefore, the number of the database files, N, would need to be large enough, and in practice usually is, for the fraction of the cache memory assigned for the encryption keys to be negligible.

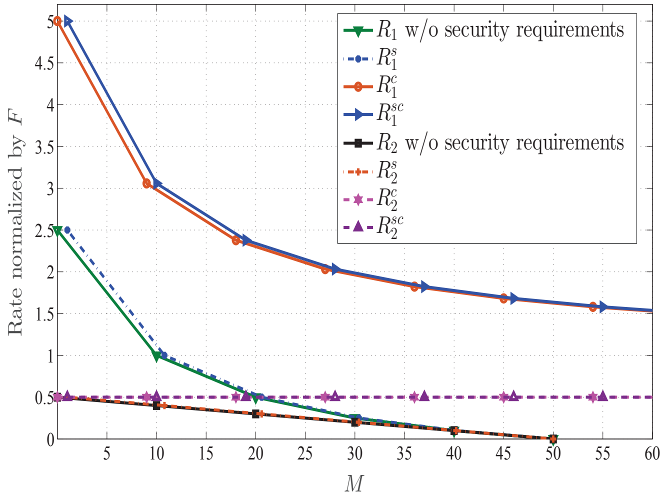

Fig. 2: Comparison between the required transmission rates under different system requirements for $N=50$, $K=15$, $h=6$ and $r=2$.

VI. Discussion

In Fig. 2, we compare the achievable rates under different requirements. First, we note that thanks to the unicast nature of communications between the relay nodes and the end users, in all cases we can achieve the lower bound on R_2, i.e., there is no overhead in the transmissions over the second hop. In particular, under secure delivery, each user caches a fraction $\frac{M-1}{N-1}$ of each file, and the total data received by any end user under secure delivery equals $(1-\frac{M-1}{N-1})F$, which is the minimum number of bits required to reconstruct the requested file. Similarly, in the two remaining scenarios, we know from the result in reference [3] that the minimum number of bits required by each user to be able to recover its requested file is F, and our achievable schemes achieve this lower bound.

We remark that the achievable scheme in Section III, can be used in the case of untrusted relays, i.e., honest-but-curious relay nodes [10] [11]. An additional step is needed during the cache placement phase to ensure that the encryption keys are not compromised by a relay node. The server should encode the cache contents, $Z_{\mathcal{V}}$, using a proper secret sharing scheme, and transmit the resulting shares to user $U_{\mathcal{V}}$ via the relay nodes indexed by $\mathcal{V}$ such that the shares pass via each relay node cannot reveal any information about the cache contents. Such scheme is applicable even in the case of colluding relays as long as the number of colluding relays is less than r.

Another observation is that under secure coded caching requirement only (Section IV), we do not need to use keys in order to ensure the secure coded caching requirement, in contrast with the general scheme in [3]. This follows from the network structure, as the relay nodes unicast the signals to each of the end users. In particular, the received signals by user $U_{\mathcal{V}}$ are formed by combinations of the shares in its memory and "fresh" shares of the requested file. Thus, at the end of communications, it has $r\binom{\hat{K}}{t}$ shares of the file $W_{d_{\mathcal{V}}}$, and only $r\binom{\hat{K}-1}{t-1}$ shares of the remaining files, i.e., the secure coded caching requirement is satisfied, without a need for encryption. In addition, for the case where $M=0$, i.e., no cache memory at the end users, secure coded caching is possible via routing, unlike the case in [3], where M must be at least 1.

From Fig. 2, we observe that the cost of imposing secure delivery requirement to the system is negligible for realistic system parameters. In particular, the gap between the achievable rates of the system without security and the system with secure delivery requirement vanishes as M increases. Same observation holds for the gap between the rates with secure coded caching and those with secure coded caching and secure delivery.

VII. Conclusions

In this work, we have investigated the performance of two-hop cache-aided networks under different security requirements. In particular, we have studied a network that satisfies the resolvability property, where a single server is connected to a set of end users, equipped with cache memories, via a set of relay nodes. We have studied the network with secure delivery constraints, secure coded caching constraints, as well as both secure delivery and secure coded caching constraints. We have provided achievability schemes for each of these requirements where the cache placement and delivery phases are handled accordingly. The proposed schemes utilize secret sharing and one-time padding. Our work demonstrates the impact of the network topology and strategies applied at the intermediate nodes on the performance of the network under different security requirements. Future directions include investigating networks that do not satisfy the resolvability criteria, and networks where the relay nodes have cache memories.

References

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

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

[3] N. K. V. Ravindrakumar, P. Panda and V. Prabhakaran, "Fundametal limits of secretive coded caching," in *IEEE International Symposium on Information Theory (ISIT)*, 2016, pp. 425–429.

[4] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. Diggavi, "Hierarchical coded caching," in *IEEE International Symposium on Information Theory (ISIT)*, 2014, pp. 2142–2146.

[5] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Caching in combination networks," in *49th Asilomar Conference on Signals, Systems and Computers*, 2015, pp. 1269–1273.

[6] M. Ji, M. F. Wong, A. M. Tulino, J. Llorca, G. Caire, M. Effros, and M. Langberg, "On the fundamental limits of caching in combination networks," in *16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2015, pp. 695–699.

[7] N. K. V. Ravindrakumar, P. Panda and V. Prabhakaran, "Coded caching for networks with the resolvability property," in *IEEE International Symposium on Information Theory (ISIT)*, 2016, pp. 420–424.

[8] M. A. Maddah-Ali and U. Niesen, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Trans. on Networking*, vol. 23, no. 4, pp. 1029–1040, 2015.

[9] I. B. D. R. Cramer and J. B. Nielsen, *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.

[10] X. He and A. Yener, "Cooperation with an untrusted relay: A secrecy perspective," *IEEE Trans. Info. Theory*, vol. 56, no. 8, pp. 3807–3827, 2010.

[11] A. A. Zewail, M. Nafea, and A. Yener, "Multi-terminal networks with an untrusted relay," in *52 Annual Allerton Conf. On Communication, Control and Computing*, 2014, pp. 895–902.