Cache-Aided Combination Networks with Secrecy Guarantees

Ahmed A. Zewail and Aylin Yener

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

***Abstract*—In this paper, we study a cache-aided combination network, where a layer of relay nodes connects a server to a set of end users, under three different secrecy requirements. Both the relays and the end users are equipped with caches. First, we consider secure delivery where we require that an external entity must not gain any information about the database files by observing the transmitted signals over the network links. Second, we consider secure caching where we require that an end user must not be able to obtain any information about a file that he did not request. Last, we consider both of these requirements simultaneously. We jointly optimize the cache placement and delivery phases in order to minimize the delivery load over each of the two hops, and demonstrate the impact of the network topology on the system performance under these secrecy requirements.**

I. Introduction

Coded caching [1] is a potential solution to alleviate network congestion for 5G systems and beyond. During the *cache placement phase* that takes place in off-peak hours, cache memories of the network nodes are populated with functions of the data files expected to be requested in the near future. During peak periods, when the users request data contents, known as the *delivery phase*, the network load can be reduced thanks to the cached contents. In addition to the multicast network considered in [1], several network topologies with caching capabilities have been investigated. In particular, references [2]–[5] have investigated a model where multiple relay nodes connect to a server, and serve each user via unicast links. In this symmetric two-hop network, known as a *combination network*, the server is connected to a set of h relay nodes, and each end user is connected to exactly r relay nodes. References [2]–[4] have studied combination networks with caches at the end users only. In reference [5], we have introduced caches at the relay nodes in addition to those at the end users. Utilizing maximum distance separable (MDS) codes [6], we have decomposed the combination network into h virtual sub-networks, and jointly optimized the cache placement and delivery policies to provide an upper bound on the transmission rates during the delivery phase. In this paper, we will consider the same model with security guarantees.

Information security is one of the major concerns in today's communication systems. Streaming services require paid subscribers for access to their database contents. This calls for cache-aided systems that not only reduce the delivery load but also keep the content secret from unauthorized parties. In this paper, we investigate securing a combination network with caches at both relay nodes and end users under three different scenarios. In the first scenario, we consider the case where the database must be kept secret from any external eavesdropper that overhears the delivery phase, i.e., *secure delivery* [7]. In the second scenario, we consider the case where end users must not be able gain any information about the files that they did not request, i.e., *secure caching* [8] [9]. Last, we consider both requirements simultaneously. Utilizing the decomposition approach in [5], we jointly optimize the cache placement and delivery phases using one-time padding [10] and secret sharing schemes [11]. Note that in secrecy for cache-aided combination networks, previous work to date consists of our recent effort [12], where the proposed schemes are limited to resolvable combination networks with no caching relays [3].

Our study demonstrates the impact of the network topology on the system performance under secrecy requirements. In particular, we demonstrate that satisfying the *secure caching* constraint in a combination network does not require encryption keys and is possible even when the memory size of the end user is less than the file size. This is in contrast to references [8] and [9] for a multicast network and a device-to-device system, respectively. Additionally, we observe that the cost of imposing the *secure delivery* requirement is negligible in combination networks, similar to the shared multicast link [7]. Finally, we observe that the caches can help disengage the server during the delivery phase even with confidentiality requirements. For all the considered scenarios, our proposed schemes are optimal over the second hop.

The remainder of the paper is organized as follows. Section II describes the system model. In Sections III, IV and V, we detail the achievability techniques for the three secrecy scenarios. In Section VI, we discuss our results. Section VII concludes the paper.

II. System Model

A. Network Model

Consider a combination network, where a server, S, is connected to K end users via a layer of h relay nodes as illustrated in Fig. 1. In particular, each end user is connected to a distinct set of r relay nodes, $r < h$. Thus, we have $K = \binom{h}{r}$

978-1-5386-0579-0/18/$31.00 ©2018 IEEE

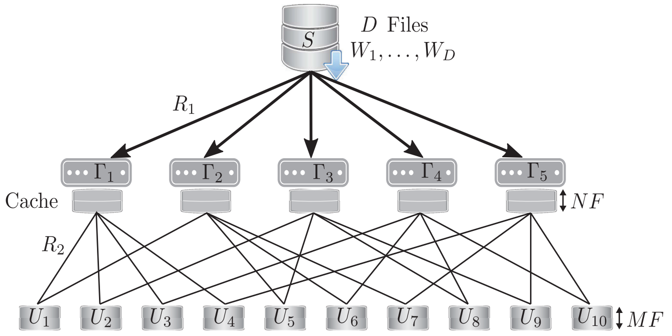

Fig. 1: A combination network with $K = 10$, $h = 5$, $r = 2$, and caches at both relays and end users.

and each relay node is connected to $\hat{K} = \binom{h-1}{r-1} = \frac{rK}{h}$ end users. Similar to references [2]–[5], all network links are *unicast* and noiseless. We denote by $\mathcal{R}=\{\Gamma_1, .., \Gamma_h\}$ the set of relay nodes, and by $\mathcal{U}=\{U_1, .., U_K\}$ the set of all end users. In addition, we define $\mathcal{N}(\Gamma_i)$ to represent the set of end users connected to Γ_i, where $|\mathcal{N}(\Gamma_i)|=\hat{K}$ for $i = 1, .., h$, and the set relay nodes connected to user k as $\mathcal{N}(U_k)$, $|\mathcal{N}(U_k)| = r$.

The function $Index(,) : (i, k) \rightarrow \{1, .., \hat{K}\}$, where $i \in \{1, .., h\}$ and $k \in \mathcal{N}(\Gamma_i)$, is defined as a function that orders the end users connected to relay node Γ_i in an ascending manner. For example, for the network in Fig. 1, $\mathcal{N}(\Gamma_2) = \{1, 5, 6, 7\}$, $\mathcal{N}(\Gamma_4) = \{3, 6, 8, 10\}$, and

$$Index(2, 1) = 1, \; Index(2, 5) = 2, \; Index(2, 6) = 3,$$
$$Index(2, 7) = 4, \; Index(4, 3) = 1, \; Index(4, 10) = 4.$$

We use the notation $[L] \triangleq \{1, .., L\}$, for a positive integer L.

B. Caching Model

The server has a database of D files, $W_1, .., W_D$, each of size F bits. The files are independent and uniformly distributed over the set $[2^F]$. We consider the case where the number of users is less than or equal to the number of files, i.e., $K \leq D$. Each relay node has cache of size NF bits, while each end user is equipped with a cache memory of size MF bits, i.e., N and M denote the normalized cache memory sizes at the relay nodes and end users, respectively. The system operates over two consecutive phases.

1) Cache Placement Phase: The server places functions of its database files in the relay nodes end users caches. The allocation is done ahead of and without the knowledge of the actual demand of the individual users.

Definition 1. *(Cache Placement): The cached contents by relay node i and user k, respectively, are given by*

$$V_i = \nu_i(W_1, W_2, .., W_D), \quad Z_k = \phi_k(W_1, W_2, .., W_D), \quad (1)$$

where $\nu_i : [2^F]^D \rightarrow [2^F]^N$ and $\phi_k : [2^F]^D \rightarrow [2^F]^M$, i.e., $H(V_i) \leq NF$ and $H(Z_k) \leq MF$. ■

2) Delivery Phase: Each user requests a file independently and randomly [1]. d_k denotes the index of the requested file by user k, i.e., $d_k \in \{1, 2, .., D\}$, and $\boldsymbol{d}$ represents the demand vector of all users. The server responds to users' requests by transmitting signals to the relay nodes. Then, each relay transmits *unicast* signals to its connected end users. From the r received signals and Z_k, user k must be able to decode W_{d_k}.

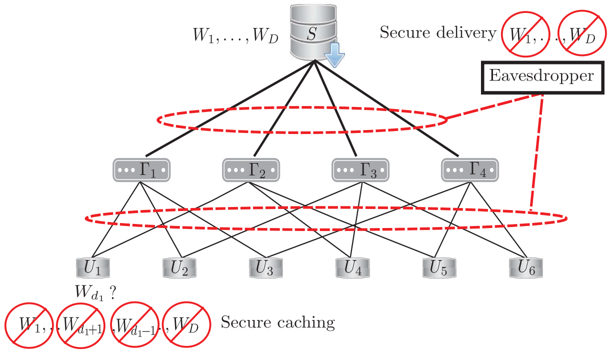

Fig. 2: A combination network with secrecy requirements.

Definition 2. *(Coded Delivery): The mapping from the database files and the demand vector $\boldsymbol{d}$ into the transmitted signal by the server to Γ_i is given by the encoding function*

$$X_{i,\boldsymbol{d}} = \psi_i(W_1, .., W_D, \boldsymbol{d}), \qquad i = 1, 2, .., h, \quad (2)$$

where $\psi_i : [2^F]^D \times [D]^K \rightarrow [2^F]^{R_1}$, and R_1 is the rate, normalized by the file size, F, of the transmitted signal from the server to each relay node. The transmitted signal from Γ_i to user $k \in \mathcal{N}(\Gamma_i)$, is given by the encoding function

$$Y_{i,\boldsymbol{d},k} = \varphi_k(X_{i,\boldsymbol{d}}, V_i, \boldsymbol{d}), \quad (3)$$

where $\varphi_k : [2^F]^{R_1} \times [2^F]^N \times [D]^K \rightarrow [2^F]^{R_2}$, and R_2 is the normalized rate of the transmitted signal from a relay node to a connected end user. User k recovers its request by

$$\hat{W}_{d_k} = \mu_k(Z_k, \boldsymbol{d}, \{Y_{i,\boldsymbol{d},k} : i \in \mathcal{N}(U_k)\}), \quad (4)$$

where $\mu_k : [2^F]^M \times [D]^K \times [2^F]^{rR_2} \rightarrow [2^F]$ is the decoding function. ■

We require that each end user k recover its requested file reliably, i.e., for any $\epsilon > 0$,

$$\max_{\boldsymbol{d},k} P(\hat{W}_{d_k} \neq W_{d_k}) < \epsilon. \quad (5)$$

In the following sections, we treat the system under each of the three different secrecy requirements. We provide brief descriptions of the schemes and results, noting that full details can be found in [13].

III. Coded Caching with Secure Delivery

In this section, we investigate the system with *secure delivery* requirement. This means that an external eavesdropper who overhears the delivery phase must not gain any information about the files, see Fig. 2. More formally, for any $\delta > 0$, we have the secrecy constraint

$$I(\mathcal{X}, \mathcal{Y}; W_1, ..W_D) < \delta, \quad (6)$$

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

In order to satisfy (6), we place keys in the network caches during the cache placement phase. These same keys are used to encrypt, i.e., one-time pad [10], the transmitted signals during the delivery phase.

A. Cache Placement Phase

We start by providing a scheme for $M=1+\frac{t_2(D-1)}{\hat{K}}+\frac{(t_1-t_2)r(D-1)N}{\hat{K}(N+\hat{K}-t_1)}$, where $t_1,t_2\in\{0,1,..,\hat{K}\}$ and $\frac{t_1}{\hat{K}}\leq\frac{N}{D+\hat{K}-t_1}$. Other values of M are achievable by memory sharing [1]. First, the server encodes each file, W_n, using an (h,r) MDS code to obtain the encoded symbols $\{f_n^i: i\in[h]\}$ [6]. Each symbol has a size of $\frac{F}{r}$ bits and any r symbols are sufficient to reconstruct the file. Then, we divide each encoded symbol into two parts, $f_n^{i,1}$ with size $\frac{NF}{D+\hat{K}-t_1}$ bits and $f_n^{i,2}$ with size $\frac{F}{r}-\frac{NF}{D+\hat{K}-t_1}$ bits. Next, the server places $f_n^{i,1}$ in the cache of relay node Γ_i. User k, with $k\in\mathcal{N}(\Gamma_i)$, caches a random fraction of $\frac{t_1}{\hat{K}}$ bits from $f_n^{i,1}$, which we denote by $f_{n,k}^{i,1}$. $f_n^{i,2}$ is divided into $\binom{\hat{K}}{t_2}$ disjoint pieces each of which is denoted by $f_{n,\mathcal{T}}^{i,2}$, where $\mathcal{T}\subseteq[\hat{K}]$, and $|\mathcal{T}|=t_2$. The server allocates the pieces $\{f_{n,\mathcal{T}}^{i,2},\ \forall n\}$ in the cache memory of user k if $k\in\mathcal{N}(\Gamma_i)$ and $Index(i,k)\in\mathcal{T}$. The size of each piece is $\frac{\frac{F}{r}-\frac{NF}{D+\hat{K}-t_1}}{\binom{\hat{K}}{t_2}}F$ bits.

To facilitate secure delivery, the server generates $h\binom{\hat{K}}{t_2+1}$ independent keys. Each key is uniformly distributed with length $\frac{\frac{F}{r}-\frac{NF}{D+\hat{K}-t_1}}{\binom{\hat{K}}{t_2}}$ bits and denoted by $K_{\mathcal{T}_K}^i$, where $i=1,..,h$, and $\mathcal{T}_K\subseteq[\hat{K}],|\mathcal{T}_K|=t_2+1$. User k stores the keys $K_{\mathcal{T}_K}^i$, $\forall i\in\mathcal{N}(U_k)$, whenever $Index(i,k)\in\mathcal{T}_K$. In addition, the sever generates the random keys K_l^i each of length $\frac{NF(\hat{K}-t_1)}{(D+\hat{K}-t_1)\hat{K}}$ bits, for $i=1,..,h$ and $l=1,..,\hat{K}$. K_l^i will be cached by relay i and user k with $Index(i,k)=l$. These allocations satisfy the memory capacity constraints [13].

B. Coded Delivery Phase

After announcing the demand vector $\boldsymbol{d}$ in the network, for each relay Γ_i, at each transmission instance, we consider $\mathcal{S}\subseteq[\hat{K}]$, where $|\mathcal{S}|=t_2+1$. For each $\mathcal{S}$, the server sends to Γ_i,

$$X_{i,\boldsymbol{d}}^{\mathcal{S}}=K_{\mathcal{S}}^i \bigoplus_{\{k:k\in\mathcal{N}(\Gamma_i),\ Index(i,k)\in\mathcal{S}\}} f_{d_k,\mathcal{S}\setminus\{Index(i,k)\}}^i. \tag{7}$$

In total, the server transmits the signal $X_{i,\boldsymbol{d}}=\bigcup_{\mathcal{S}\subseteq[\hat{K}]:|\mathcal{S}|=t_2+1}\{X_{i,\boldsymbol{d}}^{\mathcal{S}}\}$ to Γ_i. Then, Γ_i forwards the signal $X_{i,\boldsymbol{d}}^{\mathcal{S}}$ to user k whenever $Index(i,k)\in\mathcal{S}$. In addition, the relay Γ_i sends $\{f_{d_k}^{i,1}\setminus f_{d_k,k}^{i,1}\}$ to user k encrypted by the key K_l^i such that $Index(i,k)=l$. Thus, we get

$$Y_{i,\boldsymbol{d},k}=\left\{K_l^i\oplus\{f_{d_k}^{i,1}\setminus f_{d_k,k}^{i,1}\}\right\}\bigcup_{\mathcal{S}:Index(i,k)\in\mathcal{S}}\{X_{i,\boldsymbol{d}}^{\mathcal{S}}\}. \tag{8}$$

During decoding, user k decrypts its received signals using the cached keys. Then, it recovers the pieces $\left\{f_{d_k,\mathcal{T}}^{i,2}:\mathcal{T}\subseteq[\hat{K}]\setminus\{Index(i,k)\}\right\}$ from the signals received from Γ_i, utilizing its cached contents. Thus, user k recovers $f_{d_k}^{i,2}$. In addition, user k directly gets $f_{d_k}^{i,1}$ from its the signal transmitted by relay i. Thus, it can obtain $f_{d_k}^i$. Since, user k receives signals from r relay nodes, it decodes the encoded symbols $f_{d_k}^i$, $\forall i\in\mathcal{N}(U_k)$, and is able to successfully reconstruct its requested file W_{d_k}.

Remark 1. *The server sends $h\binom{\hat{K}}{t_2+1}$ signals, each of which is encrypted using a one-time pad whose length equals to the length of each transmission, thus the prefect secrecy is ensured [10]. Observing these signals without knowing the encryption keys does not reveal any information about the database files [10]. The same applies for the messages transmitted by the relay nodes. Thus, (6) is satisfied.* ■

C. Secure Delivery Rates

We denote the secure delivery rates in the first and second hop with R_1^s and R_2^s, respectively. Each relay node is responsible for $\binom{\hat{K}}{t_2+1}$ transmissions, each of length $\frac{|f_n^{i,2}|}{\binom{\hat{K}}{t_2}}$, thus

$$R_1^sF=\frac{\binom{\hat{K}}{t_2+1}}{\binom{\hat{K}}{t_2}}|f_n^{i,2}|=\frac{\hat{K}-t_2}{(t_2+1)}\left(\frac{F}{r}-\frac{NF}{D+\hat{K}-t_1}\right). \tag{9}$$

Over the second hop, Γ_i forwards $\binom{\hat{K}-1}{t_2}$ from its received signals to each connected end users, in addition to transmitting a message of size $\frac{N(\hat{K}-t_1)F}{(D+\hat{K}-t_1)\hat{K}}$ bits from its cached contents to each user. Therefore, we have

$$R_2^sF=\frac{\binom{\hat{K}-1}{t_2}|f_n^{i,2}|}{\binom{\hat{K}}{t_2}}+\frac{NF(\hat{K}-t_1)}{(D+\hat{K}-t_1)\hat{K}}=\frac{F}{r}\left(1-\frac{M-1}{N-1}\right). \tag{10}$$

Consequently, we have the following theorem.

Theorem 1. *The normalized transmission rates with secure delivery, for $N\geq 0$, $M=1+\frac{t_2(D-1)}{\hat{K}}+\frac{(t_1-t_2)r(D-1)N}{\hat{K}(N+\hat{K}-t_1)}$, $t_1,t_2\in\{0,1,..,\hat{K}\}$ and $\frac{t_1}{\hat{K}}\leq\frac{N}{D+\hat{K}-t_1}$, are upper bounded by*

$$R_1^s\leq\frac{\hat{K}-t_2}{r(t_2+1)}\left(1-\frac{Nr}{D+\hat{K}-t_1}\right), R_2^s\leq\frac{1}{r}\left(1-\frac{M-1}{D-1}\right). \tag{11}$$

In addition, the convex envelope of these points is achievable by memory sharing. ■

For the special case of no caches at the relays, i.e., $N=t_1=0$, we obtain the following upper bound on the secure delivery rates.

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

$$R_1^s\leq\frac{\hat{K}\left(1-\frac{M-1}{D-1}\right)}{r\left(\hat{K}\frac{M-1}{D-1}+1\right)},\qquad R_2^s\leq\frac{1}{r}\left(1-\frac{M-1}{D-1}\right). \tag{12}$$

In addition, the convex envelope of these points is achievable by memory sharing. ■

IV. Combination networks with secure caching

Next, we consider *secure caching*, i.e., an end user must *not* be able to obtain any information about the files that he did not request, see Fig. 2. Formally, we must have, for $\delta > 0$

$$\max_{\boldsymbol{d},k} I(\boldsymbol{W}_{-d_k}; \{Y_{i,\boldsymbol{d},k} : i \in \mathcal{N}(U_k)\}, Z_k) < \delta, \qquad (13)$$

where $\boldsymbol{W}_{-d_k} = \{W_1, .., W_N\} \backslash \{W_{d_k}\}$, i.e., the set of all files except the one requested by user k.

For achievability in this scenario, we utilize secret sharing schemes [11], [14] to ensure that no user is able to obtain information about the files from its cached contents. The basic idea of secret sharing schemes is to encode the secret in such a way that accessing a subset of shares does not suffice to reduce the uncertainty about the secret. In particular, we utilize a class of secret sharing schemes known as *non-perfect secret sharing schemes*, defined as follows.

Definition 3. *[11] [14] For a secret W with size F bits, an (m, n) non-perfect secret sharing scheme generates n shares, $S_1, S_2, ..S_n$, such that accessing any m shares does not reveal any information about the file W, i.e.,*

$$I(W; \mathcal{S}) = 0, \quad \forall \mathcal{S} \subseteq \{S_1, S_2, ..S_n\}, |\mathcal{S}| \leq m. \qquad (14)$$

Furthermore, W can be losslessly reconstructed from the n shares, i.e.,

$$H(W|S_1, S_2, .., S_n) = 0. \qquad (15)$$

■

For large enough F, an (m, n) secret sharing scheme exists with shares of size equal to $\frac{F}{n-m}$ bits [11], [14].

A. Cache Placement Phase

As a first step, the server encodes each file using an (h, r) MDS code. We denote by f_n^i the resulting encoded symbols, where n is the file index and $i = 1, 2, .., h$. For $M = \frac{tD}{\hat{K}-t}(1 - \frac{Nr}{D})$ and $t \in \{0, 1, .., \hat{K}-1\}$, we divide each encoded symbol into two parts, $f_n^{i,1}$ with size $\frac{NF}{D}$ bits and $f_n^{i,2}$ with size $\frac{F}{r} - \frac{NF}{D}$ bits. The parts $\{f_n^{i,1} : \forall n\}$ will be cached by Γ_i and will not be cached by any end user. $f_n^{i,2}$ is encoded using an $\left(\binom{\hat{K}-1}{t-1}, \binom{\hat{K}}{t}\right)$ secret sharing scheme [11], [14]. The resulting shares are denoted by $S_{n,\mathcal{T}}^i$, where n is the file index i.e., $n \in \{1, .., N\}$, i is the index of the encoded symbol, i.e., $i = 1, .., h$, and $\mathcal{T} \subseteq [\hat{K}], |\mathcal{T}| = t$. Each share has size

$$F_s = \frac{\frac{F}{r} - \frac{NF}{D}}{\binom{\hat{K}}{t} - \binom{\hat{K}-1}{t-1}} = \frac{t\left(1 - \frac{Nr}{D}\right)}{r(\hat{K} - t)\binom{\hat{K}-1}{t-1}} F \text{ bits}. \qquad (16)$$

The server allocates the shares $S_{n,\mathcal{T}}^i, \forall n$ in the cache of user k whenever $i \in \mathcal{N}(U_k)$ and $Index(i, k) \in \mathcal{T}$. Such allocations satisfy the memory capacity constraints [13], and gives $t = \frac{KrM}{h(D+M)}$.

B. Coded Delivery Phase

Each user requests a file from the server. First, we focus on the transmission from the server to Γ_i. At each transmission instance, we consider $\mathcal{S} \subseteq [\hat{K}]$, where $|\mathcal{S}| = t + 1$. For each $\mathcal{S}$, the server sends to Γ_i, the signal

$$X_{i,\boldsymbol{d}}^{\mathcal{S}} = \bigoplus_{\{k: k \in \mathcal{N}(\Gamma_i),\ Index(i,k) \in \mathcal{S}\}} S_{d_k, \mathcal{S} \backslash \{Index(i,k)\}}^i. \qquad (17)$$

In total, the server transmits to Γ_i, the signal $X_{i,\boldsymbol{d}} = \bigcup_{\mathcal{S} \subseteq [\hat{K}]: |\mathcal{S}| = t+1} \{X_{i,\boldsymbol{d}}^{\mathcal{S}}\}$. Then, Γ_i forwards $X_{i,\boldsymbol{d}}^{\mathcal{S}}$ to user k whenever $Index(i, k) \in \mathcal{S}$. In addition, Γ_i sends directly $f_{d_k}^{i,1}$ to user k. Therefore, we have

$$Y_{i,\boldsymbol{d},k} = \{f_{d_k}^{i,1}\} \bigcup_{\mathcal{S}: Index(i,k) \in \mathcal{S}} \{X_{i,\boldsymbol{d}}^{\mathcal{S}}\}. \qquad (18)$$

User k can recover $\{S_{d_k,\mathcal{T}}^i : \mathcal{T} \subseteq [\hat{K}] \setminus \{Index(i, k)\}, |\mathcal{T}| = t\}$ from the signals received from Γ_i, using its cache's contents. Adding these shares to the cached ones, i.e., $S_{d_k,\mathcal{T}}^i$ with $Index(i, k) \in \mathcal{T}$, user k decodes $f_{d_k}^{i,2}$ from its $\binom{\hat{K}}{t}$ shares. Thus, user k obtains the encoded symbols $f_{d_k}^i$, $\forall i \in \mathcal{N}(U_k)$, and reconstructs W_{d_k}.

C. Secure Caching Rates

With the secure caching requirement, we denote the first and second hop rates by R_1^c and R_2^c, respectively. Each relay is responsible for $\binom{\hat{K}}{t+1}$ transmissions, each of length F_s bits, thus the transmission rate, in bits, from the server to each relay node is

$$R_1^c F = \frac{t\binom{\hat{K}}{t+1}\left(1 - \frac{Nr}{D}\right) F}{r(\hat{K} - t)\binom{\hat{K}-1}{t-1}} = \frac{\hat{K}\left(1 - \frac{Nr}{D}\right) F}{r(t+1)}. \qquad (19)$$

Each relay forwards $\binom{\hat{K}-1}{t}$ of these signals to each of its connected end users. In addition, each relay node sends $\frac{NF}{D}$ bits from its cache to each of these users. Therefore, we have

$$R_2^c F = \binom{\hat{K} - 1}{t} \frac{t\left(1 - \frac{Nr}{D}\right)}{r(\hat{K} - t)\binom{\hat{K}-1}{t-1}} F + \frac{NF}{D} = \frac{1}{r} F. \qquad (20)$$

Consequently, we have the following theorem.

Theorem 2. *The normalized rates with secure caching, for $0 \leq N \leq \frac{D}{r}$, $M = \frac{tD}{\hat{K}-t}(1 - \frac{Nr}{D})$, and $t \in \{0, 1, .., \hat{K}-1\}$, are upper bounded by*

$$R_1^c \leq \frac{\hat{K}(D + M - rN)}{r\left((\hat{K}+1)M + D - rN\right)} \left(1 - \frac{Nr}{D}\right), \quad R_2^c \leq \frac{1}{r}. \qquad (21)$$

The convex envelope of these points is achievable by memory sharing. ■

Remark 2. *Secret sharing schemes ensure that no user is able to reconstruct any file from its cache contents only, as the cached shares are not sufficient to reveal any information about any file. In addition, the only new information in the received signals by any end user is the shares from to its requested file. Thus, (13) is satisfied.* ■

For the special case of no relay caches, we obtain the following corollary.

Corollary 2. *The normalized rates with secure caching, for $N = 0$, $M = \frac{tD}{\hat{K}-t}$, and $t \in \{0, .., \hat{K}-1\}$, are upper bounded by*

$$R_1^c \leq \frac{\hat{K}(D+M)}{r\left((\hat{K}+1)M + D\right)}, \qquad R_2^c \leq \frac{1}{r}. \qquad (22)$$

The convex envelope of these points is achievable by memory sharing. ■

V. Simultaneous secure caching and delivery

In this section, we consider both secure caching and secure delivery simultaneously. In the proposed achievability scheme, we utilize both one-time pads and secret sharing.

A. Cache Placement Phase

For $M= 1+\frac{tD}{\hat{K}-t}(1-\frac{rN}{D+\hat{K}})$, and $t \in \{0,1,..,\hat{K}-1\}$, after encoding each file using an (h,r) MDS code, we divide each encoded symbol into two parts, $f_n^{i,1}$ with size $\frac{NF}{D+\hat{K}}$ bits and $f_n^{i,2}$ with size $\frac{F}{r}-\frac{NF}{D+\hat{K}}$ bits. Only Γ_i caches the parts $\{f_n^{i,1} : \forall n\}$. We encode $f_n^{i,2}$ using an $\left(\binom{\hat{K}-1}{t-1},\binom{\hat{K}}{t}\right)$ secret sharing scheme from [11], [14]. The resulting shares are denoted by $S_{n,\mathcal{T}}^i$, where n is the file index, i is the index of the encoded symbol and $\mathcal{T} \subseteq [\hat{K}], |\mathcal{T}| = t$. Each share has size

$$F_s = \frac{\frac{F}{r}-\frac{NF}{D+\hat{K}}}{\binom{\hat{K}}{t}-\binom{\hat{K}-1}{t-1}} = \frac{t\left(1-\frac{Nr}{D+\hat{K}}\right)}{r(\hat{K}-t)\binom{\hat{K}-1}{t-1}} F \text{ bits.} \tag{23}$$

The server places the shares $S_{n,\mathcal{T}}^i$, $\forall n$ in the cache of user k whenever $i \in \mathcal{N}(U_k)$ and $Index(i,k) \in \mathcal{T}$.

The server generates $h\binom{\hat{K}}{t+1}$ independent keys such that each key is uniformly distributed over the set $[2^{F_s}]$ with length F_s bits and is denoted by $K_{\mathcal{T}_K}^i$, where $i = 1,..,h$, and $\mathcal{T}_K \subseteq [\hat{K}], |\mathcal{T}_K| = t+1$. User k caches the keys $K_{\mathcal{T}_K}^i$, $\forall i \in \mathcal{N}(U_k)$, whenever $Index(i,k) \in \mathcal{T}_K$. Furthermore, the sever generates the random keys K_l^i each of length $\frac{NF}{D+\hat{K}}$ bits, for $i = 1,..,h$ and $l = 1,..,\hat{K}$, to be cached by relay i and user k with $Index(i,k) = l$. This scheme satisfies the memory constraints [13], and gives $t = \frac{\hat{K}(M-1)(D+\hat{K})}{(D+\hat{K})(M+D-1)+rND}$.

B. Coded Delivery Phase

The delivery phase begins with announcing the demand vector to all network nodes. For Γ_i, at each transmission instance, we consider a $\mathcal{S} \subseteq [\hat{K}]$, where $|\mathcal{S}| = t+1$. For each $\mathcal{S}$, the server transmits to Γ_i, the following signal

$$X_{i,\boldsymbol{d}}^{\mathcal{S}} = K_{\mathcal{S}}^i \bigoplus_{\{k:k\in\mathcal{N}(\Gamma_i),\ Index(i,k)\in\mathcal{S}\}} S_{d_k,\mathcal{S}\setminus\{Index(i,k)\}}^i. \tag{24}$$

Thus, the server transmits to Γ_i, the signal $X_{i,\boldsymbol{d}} = \bigcup_{\mathcal{S}\subseteq[\hat{K}]:|\mathcal{S}|=t+1}\{X_{i,\boldsymbol{d}}^{\mathcal{S}}\}$. Then, Γ_i forwards the signal $X_{i,\boldsymbol{d}}^{\mathcal{S}}$ to user k whenever $Index(i,k) \in \mathcal{S}$. In addition, Γ_i sends $f_{d_k}^{i,1}$ encrypted by K_l^i to user k such that $Index(i,k) = l$. After decrypting the received signals, user k get $f_{d_k}^{i,1}$ and can extract the set of shares $\{S_{d_k,\mathcal{T}}^i : \mathcal{T} \subseteq [\hat{K}] \setminus \{Index(i,k)\}, |\mathcal{T}| = t\}$ from the signals received from Γ_i. These shares in addition to the ones in its cache, i.e., $S_{d_k,\mathcal{T}}^i$ with $Index(i,k) \in \mathcal{T}$, allow user k to decode $f_{d_k}^{i,2}$ from its $\binom{\hat{K}}{t}$ shares. Since, user k receives signals from r different relay nodes, it obtains $\{f_{d_k}^i$, $\forall i \in \mathcal{N}(U_k)\}$, then decodes W_{d_k}.

C. Secure Caching and Secure Delivery Rates

We refer to the first and second hop rates as R_1^{sc} and R_2^{sc}, respectively. The server sends to each relay node $\binom{\hat{K}}{t+1}$ signals, each of length F_s bits, thus we have

$$R_1^{sc}F = \frac{\binom{\hat{K}}{t+1}t\left(1-\frac{Nr}{D+\hat{K}}\right)F}{r(\hat{K}-t)\binom{\hat{K}-1}{t-1}} = \frac{\hat{K}F}{r(t+1)}\left(1-\frac{Nr}{D+\hat{K}}\right). \tag{25}$$

Over second hop, each relay node is responsible for forwarding $\binom{\hat{K}-1}{t}$ from its received signals to each of its connected end users, in addition to delivering $\frac{NF}{D+\hat{K}}$ bits from its cache, thus

$$R_2^{sc}F = \binom{\hat{K}-1}{t}\frac{t\left(1-\frac{Nr}{D+\hat{K}}\right)F}{r(\hat{K}-t)\binom{\hat{K}-1}{t-1}} + \frac{NF}{D+\hat{K}} = \frac{1}{r}F. \tag{26}$$

Therefore, we can obtain the following theorem.

Theorem 3. *Under secure delivery and secure caching requirements, for $0 \leq N \leq \frac{D+\hat{K}}{r}$, $M= 1+\frac{tD}{\hat{K}-t}(1-\frac{rN}{D+\hat{K}})$, and $t \in \{0,1,..,\hat{K}–1\}$, the rates are upper bounded by*

$$R_1^{sc} \leq \frac{\hat{K}\left(rND+(D+\hat{K})(M+D-1)\right)\left(1-\frac{Nr}{D+\hat{K}}\right)}{r\left(rND+(D+\hat{K})[D+(M-1)(\hat{K}+1)]\right)},$$
$$R_2^{sc} \leq \frac{1}{r}. \tag{27}$$

In addition, the convex envelope of these points is achievable by memory sharing. ■

When there are no caches at the relays, we obtain

Corollary 3. *Under secure delivery and secure caching requirements, for $N = 0$, $M=\frac{tD}{\hat{K}-t}+1$, and $t \in \{0,1,..,\hat{K}–1\}$, the transmission rates are upper bounded by*

$$R_1^{sc} \leq \frac{\hat{K}(D+M-1)}{r\left((\hat{K}+1)(M-1)+D\right)}, \qquad R_2^{sc} \leq \frac{1}{r}. \tag{28}$$

In addition, the convex envelope of these points is achievable by memory sharing. ■

Remark 3. *Corollaries 1-3 effectively generalize our previous results in [12] that were limited to resolvable networks, to any combination network, i.e., we have shown the achievability of the rates in [12] for any combination network.* ■

VI. Discussion and Numerical Results

A. Secrecy Cost

In Fig. 3, we compare the achievable rates under different secrecy scenarios. We observe that the cost of imposing secure delivery is *negligible* for realistic system parameters. As M increases, the gap between the achievable rates without secrecy and those with secure delivery requirement vanishes. Same observation holds for secure caching as well as simultaneous secure caching and secure delivery.

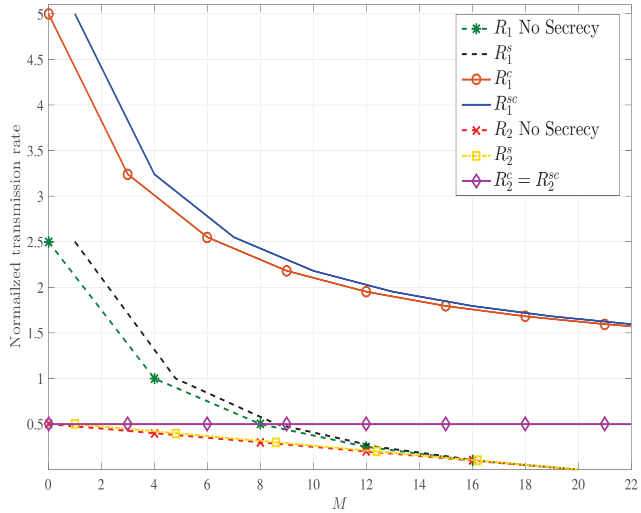

Fig. 3: Rates under different system requirements for $N=0$, $D=20$, $K=15$, $h=6$ and $r=2$.

B. Optimality over the Second Hop

The achievable rates over the second hop are optimal, i.e., achieve the minimum delivery load per relay. 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 $\left(1-\frac{M-1}{N-1}\right)F$, which is the minimum number of bits required to reconstruct the requested file [7]. Similarly, in the two remaining scenarios, from the lower bounds in [8] and [9], it is known that, under secure caching requirement, the number bits received by any user is lower bounded by the file size. Thus, the rates over the second hop, R_2^c and R_2^{sc}, in (21) and (27), respectively, are optimal.

C. Impact of Network Topology

Under secure caching requirement only, we do not need to use keys in order to ensure the secure caching requirement, in contrast with the general schemes in references [8] and [9]. 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 k are formed by combinations of the shares in its memory and "fresh" shares of the requested file. Thus, at the end of communications, user k has $r\binom{\hat{K}}{t}$ shares of W_{d_k}, and only $r\binom{\hat{K}-1}{t-1}$ shares of the remaining files, i.e., the secure caching requirement is satisfied, without the need to encrypt. In addition, for the case where $M=0$, i.e., no cache memory at the end users, secure caching is possible via routing, unlike the case in [8], where M must be at least 1.

D. Disengaging the Server from the Delivery Phase

One of the main advantages of caching relays is the ability to keep the server silent during the delivery phase while satisfying the users requests [5]. Without secrecy requirements in [5], we have shown that whenever $M+rN \geq D$, i.e., the memory of each user and its connected relay nodes is sufficient to store the whole library, all the end users' requests can be satisfied while the sever is silent during the delivery phase. Under the secure delivery requirement, we need $M+rN\left(\frac{D-1}{D+\hat{K}}\right) \geq D$ in order to disengage the server from the delivery process because the transmission from the relay nodes to the end users must be protected by shared keys between them. On the other hand, under the secure caching requirement, to achieve zero rate over the first hop, we need $rN \geq D$, where we distribute the library over the relays' caches and the unicast nature of the network links ensures confidentiality. In this case, there is no need to utilize the cache of the end user. When the system requires both secure caching and secure delivery, we need $rN \geq D+\hat{K}$ and $M \geq 1$ to achieve $R_1^{sc}=0$.

VII. Conclusion

In this work, we have investigated combination networks with caches at both relay nodes and end users under secure delivery constraints, secure caching constraints, as well as both secure delivery and secure caching constraints. We have provided achievability schemes, for each of these requirements, by jointly optimizing the cache placement and delivery phases, utilizing one-time padding and secret sharing schemes. We have illustrated the impact of the network structure and relaying on the system performance after imposing different secrecy constraints. Furthermore, we have shown that the caches at the relays in addition to the ones at the end users can completely replace the server during the delivery phase.

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

[3] L. Tang and A. Ramamoorthy, "Coded caching for networks with the resolvability property," in *IEEE International Symposium on Information Theory (ISIT)*, 2016.

[4] K. Wan, M. Ji, P. Piantanida, and D. Tuninetti, "Caching in combination networks: Novel multicast message generation and delivery by leveraging the network topology," *arXiv:1710.06752*, 2017.

[5] A. A. Zewail and A. Yener, "Coded caching for combination networks with cache-aided relays," in *IEEE International Symposium on Information Theory (ISIT)*, 2017.

[6] S. Lin and D. J. Costello, *Error control coding*. Pearson Education India, 2004.

[7] 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.

[8] V. Ravindrakumar, P. Panda, N. Karamchandani, and V. Prabhakaran, "Fundamental limits of secretive coded caching," in *IEEE International Symposium on Information Theory (ISIT)*, 2016.

[9] A. A. Zewail and A. Yener, "Fundamental limits of secure device-to-device coded caching," in *50th Asilomar Conference on Signals, Systems and Computers*. IEEE, 2016.

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

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

[12] A. A. Zewail and A. Yener, "Coded caching for resolvable networks with security requirements," in *the 3rd Workshop on Physical-Layer Methods for Wireless Security, CNS*, 2016.

[13] ——, "Combination networks with or without secrecy constraints: The impact of caching relays," *arXiv:1712.04930*, 2017.

[14] G. R. Blakley and C. Meadows, "Security of ramp schemes," in *Workshop on the Theory and Application of Cryptographic Techniques*. Springer, 1984, pp. 242–268.