Untrusted Caches in Two-layer Networks

Ahmed A. Zewail and Aylin Yener

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

***Abstract*—This work considers a network consisting of a server and a layer of relay nodes equipped with cache memories which aim to deliver content to end nodes that also have cache memories. The server and the end nodes consider the intermediate relay caches to be *untrusted* with the content. As a result, the server must design strategies to place content in relay caches not only to serve end users, but also to ensure that any a subset of them, even when colluding, cannot gain any information about the contents of the server database. The end users randomly connect to a subset of these untrusted caches at the beginning of the delivery phase via multicast links. For this network model, a coded caching scheme is developed by jointly optimizing the cache placement and delivery phases using secure regenerating codes. In addition, the scheme is extended to the setup of combination networks with untrusted relays, where the untrusted relays are connected to the end users via unicast links. The study highlights the benefits of cooperating with untrusted caches by designing the end users' caches to provide multicast opportunities in order to minimize the delivery load.**

I. Introduction

Caching content alleviates network congestion in modern communications systems. Recent advances are in the direction of coded caching, where during off-peak hours, functions of files are placed in caches of users (placement phase), followed by delivery in peak traffic hours, designed to capitalize on cache contents, benefiting multiple users simultaneously (delivery phase). The fundamental trade-off between the cache size and delivery load has been studied for different network models, see for example [1]–[4] and references therein.

Distributing content over the network nodes raises fundamental concerns about the data confidentiality and privacy. In addition to external unauthorized nodes (eavesdroppers) from whom the information needs to be kept confidential, some legitimate nodes may be a part of a public network or may be shared between subscribers of different services. Storing content destined for others by network nodes significantly improves network performance, but information unintended to these nodes must also not be decodable by these nodes. These concerns call for designing caching schemes that take explicitly into account the secrecy requirements of the system in addition to reducing the delivery load. Cache-aided systems under different secrecy requirements have been investigated in recent years [3], [5]–[8]. In references [3], [5], [6], [8], the requirement is that an external eavesdropper must not gain any information about the database files from overhearing the delivery phase signals over the network links, known as *secure delivery*. In references [3], [6], [7], it is required that an end user must not be able to obtain any information about a file that it did not request, known as *secure caching*.

In this paper, we consider a layered network consisting of a server, h intermediate relay nodes with cache memories and K end users with cache memories. During placement, the server places content in the cache memories of the intermediate nodes and in the cache memories of the end users. The server must design placement for all the caches without the knowledge of the actual demand of the users. Additionally, the connections to intermediate relay nodes from the end users are initiated only at the delivery phase, i.e., the server must design cache placement without knowing the actual network connectivity during delivery. At the beginning of the delivery phase, each end user connects to a set of r intermediate nodes at random and requests one of the database contents. Each intermediate relay then responds to its connected users' requests by sending a multicast signal over a noiseless error-free shared link. Such a communication model has been considered in [4] without security concerns. Here, we consider that the intermediate relay nodes are *untrusted* and the files must not be decodable by any of them. We further consider that any set of l, $l < r$, intermediate nodes can collude to share cached contents with each other. We refer to the set of intermediate nodes as *untrusted caches* analogous to *untrusted relays* that are introduced to information theoretic security in [9]. Like in [9], the caching relay nodes are assumed to be *honest-but-curious*, i.e., they follow the network protocols to provide the users with their requested files, however, they are unauthenticated, e.g., public access points, and the files (content) must be kept secret from them. The end user caches are considered to be trusted. The setting is described in Section II.

We develop a coded caching scheme by jointly optimizing the cache placement and the delivery phases in Section III. The contents of the untrusted caches are designed using secure regenerating codes [10], while the end users store a subset of the subfiles of the library in addition to a subset of random symbols that are used to generate the contents of the untrusted caches [10]. The key idea is to allow the end users to regenerate some of the contents at their connected untrusted caches in order to be able to benefit from the multicast signals transmitted from these untrusted caches. Section IV provides the performance of the proposed scheme. In Section V, we extend our scheme to combination networks with cache-aided

978-1-5386-9291-2/19/$31.00

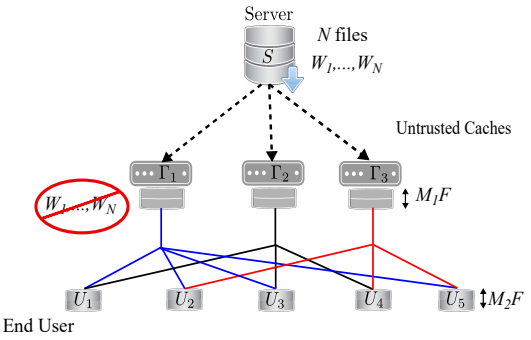

Fig. 1: A network with untrusted caches and random connectivity $K=5$, $h=3$ and $r=2$.

relays [3] with the assumption that any l relays can share their cached contents aiming to decode the database files. This section also includes discussion of insights of the work and is followed by conclusions in Section VI.

Notation: Matrices are represented by boldface letters, $\oplus$ refers to the binary XOR operation, $|W|$ denotes cardinality of W, $[K] \triangleq \{1,\dots,K\}$, and ϕ denotes the empty set.

II. System Model

Consider a network, where a main server, S, is connected to K end users, $U_1,...,U_K$, via a set of h intermediate nodes, $\Gamma_1,...,\Gamma_h$, as illustrated in Fig. 1. The server has a library of N files, $W_1,..,W_N$, each with size F bits. Each intermediate node is equipped with a cache memory of size M_1F bits, while each end user has a cache memory of size M_2F bits, i.e., M_1 and M_2 represent the normalized cache sizes. The system operates over two phases.

1) Cache Placement Phase: The server allocates functions of its database in the intermediate nodes' and end users' cache memories. These allocations are designed, without the knowledge of the actual demands nor the network connectivity in the delivery phase, subject to the memory capacity constraints.

Definition 1. *(Cache Placement): The contents of the cache memory at node Γ_j are given by*

$$V_j = \lambda_j(W_1, W_2, .., W_N), \tag{1}$$

where $\lambda_j : [2^F]^N \to [2^F]^{M_1}$, such that $H(V_j) \leq M_1F$. The contents of the cache memory at end user k are given by

$$Z_k = \phi_k(W_1, W_2, .., W_N), \tag{2}$$

where $\phi_k : [2^F]^N \to [2^F]^{M_2}$, such that $H(Z_k) \leq M_2F$. ■

The intermediate nodes are assumed to be *honest-but-curious*. In particular, we assume that any subset of l nodes can collude by sharing their cached contents aiming to gain information about the database files. Therefore, the cached contents at these nodes must satisfy the following confidentiality constraints, for any $\epsilon > 0$,

$$I(W_1,...,W_N;\mathcal{S}) \leq \epsilon, \quad \mathcal{S} \subset \{V_1,...,V_h\}, |\mathcal{S}| \leq l. \tag{3}$$

We will refer to the intermediate nodes by *untrusted caches*.

2) Delivery Phase: Each user connects to a subset of the untrusted caches and requests a file at random. In particular, user k randomly connects to a set of r untrusted caches [4]. We define the set $\mathcal{K}_j$ to be the set of end users connected to the untrusted cache j, and $K_j \triangleq |\mathcal{K}_j|$ which is a random variable with realizations denoted by k_j. Therefore, we have $\sum_{j=1}^{h} k_j = rK$. Such realizations that represent the network connectivity will be known only at the beginning of the delivery phase. In addition, we define $\mathcal{N}(U_k)$ to be the set of untrusted caches that user k connects to, i.e., $|\mathcal{N}(U_k)| = r$. For example in the network shown in Fig. 1, $\mathcal{K}_1 = \{1,2,3,5\}$, $\mathcal{K}_2 = \{1,3,4\}$, $\mathcal{K}_3 = \{2,4,5\}$, i.e., $k_1 = 4$, $k_2 = k_3 = 3$ and $\mathcal{N}(U_1) = \{1,2\}$. Each user requests a randomly selected file [1]. We denote by d_k the index of the requested file by user k, i.e., $d_k \in \{1,2,..,N\}$, and by $\boldsymbol{d}$ the demand vector of all network users at any request instance.

The network caches' must satisfy the users' requests without the participation of the main server. Therefore, each of the untrusted caches responds by transmitting a multicast signal to its connected users over an error-free unit-capacity link. In particular, Γ_j transmits to the users in $\mathcal{K}_j$, the signal

$$X_{j,\boldsymbol{d}} = \psi_i(V_j, \boldsymbol{d}), \tag{4}$$

of length R_jF bits, i.e., R_j denotes the normalized delivery load by the untrusted cache j. From the received signals and its cached contents, user k must be able to decode its requested file, reliably. In particular, user k has a decoding function to recover its requested file,

$$\hat{W}_k = \mu_k(Z_k, \boldsymbol{d}, \{X_{j,\boldsymbol{d},k} : j \in \mathcal{N}(U_k)\}), \tag{5}$$

such that, for any $\delta > 0$, $\max_{\boldsymbol{d},k} P(\hat{W}_{d_k} \neq W_{d_k}) < \delta$.

We aim to minimize the delivery time while satisfying the secrecy constraints. Similar to [4], we consider two modes of transmission: 1) *successive* transmission, and 2) *simultaneous* transmission. Under successive transmission, the communication medium between the untrusted caches and the end users is shared via time-division multiple access, i.e., no two untrusted caches can transmit, simultaneously. Thus, the normalized delivery time is given by

$$T_{\text{suc}} = \sum_{j=1}^{h} R_j. \tag{6}$$

Under simultaneous transmission, untrusted caches can transmit simultaneously, and the end users can receive from multiple untrusted caches at the same time. Therefore, the normalized delivery time is given by

$$T_{\text{sim}} = \max_j R_j. \tag{7}$$

III. The Proposed Coded Caching Scheme

In this section, we develop the achievability scheme for the case where $M_1 = \frac{N}{r-l}$, which represents the minimum memory needed to store all the files in the relays' untrusted caches, as detailed in subsection V-B. Then, we show how the scheme can be extended for other values of M_1 in subsection

V-C. First, we describe the achievability for the M_2 in the form of $\frac{Nt}{K}(\frac{l+r}{r})$, $t = 0, 1, ..., K$, and $M_2 < N$.

A. Cache Placement Phase

As a start, we divide each file W_n into $\binom{K}{t}$ disjoint subfiles each of which is denoted by $W_{n,\mathcal{T}}$. The size of each subfile is $\frac{F}{\binom{K}{t}}$ bits. Then, we encode each subfile using an (h, r, l) secure regenerating code for the minimum storage point (MSR) [10, Section VII] for a passive eavesdropper that can access the contents of l storage units. In particular, for the subfile $W_{n,\mathcal{T}}$, we divide it into r disjoint pieces, $W^i_{n,\mathcal{T}}$ and $i = 1, ..., r$, each of which has size $\frac{F}{r\binom{K}{t}}$ bits. In addition, we generate at uniformly at random, $l|W^i_{n,\mathcal{T}}|$ bits to be used as keys in the secure coding scheme. We denote them by $U_{n,\mathcal{T}}$. Both bits from $W_{n,\mathcal{T}}$ and $U_{n,\mathcal{T}}$ are placed in the message matrix $\boldsymbol{M}_{n,\mathcal{T}}$ as illustrated in [10], which we briefly describe as follows. Let $\boldsymbol{M}_{n,\mathcal{T}}$ be in the form of

$$\boldsymbol{M}_{n,\mathcal{T}} = \begin{bmatrix} \boldsymbol{S}_1 \\ \boldsymbol{S}_2 \end{bmatrix}, \tag{8}$$

where $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ are $M_1F \times M_1F$ symmetric matrices, which are populated by the symbols of $W_{n,\mathcal{T}}$. Then, we replace the first $(lM_1F - \frac{l(l-1)}{2})$ symbols in the first l rows (and hence the first l columns) of the symmetric matrix $\boldsymbol{S}_1$ and the $\frac{l(l-1)}{2}$ symbols in the intersection of the first $(l-1)$ rows and first $(l-1)$ columns of the symmetric matrix $\boldsymbol{S}_2$, with the random symbols $U_{n,\mathcal{T}}$. The encoding process is done by multiplying the generator matrix $\boldsymbol{\Psi}$, whose construction is given in [10], with the message matrix. We assume that $\boldsymbol{\Psi}$ is known to all end users. The resulting coded symbols are denoted by $C^i_{n,\mathcal{T}}$, and $i = 1, ..., h$, each of which has size $\frac{F}{(r-l)\binom{K}{t}}$ bits. The server places $C^j_{n,\mathcal{T}}$, in the untrusted cache j, while it places $W_{n,\mathcal{T}}$ and $U_{n,\mathcal{T}}$, in the cache memory of end user k as long as $k \in \mathcal{T}$. Thus, we have

$$V_j = \left\{ C^j_{n,\mathcal{T}} : \forall n, \mathcal{T} \right\}, \tag{9}$$

$$Z_k = \left\{ W_{n,\mathcal{T}}, U_{n,\mathcal{T}} : k \in \mathcal{T},\ \forall n \right\}. \tag{10}$$

This allocation satisfies the memory constraints. In particular, the number of accumulated bits at an untrusted cache is

$$N \times \binom{K}{t} \times \frac{F}{(r-l)\binom{K}{t}} = \frac{NF}{r-l} = M_1F. \tag{11}$$

The number of accumulated bits at the end user's cache is

$$\begin{aligned} &N \times \binom{K-1}{t-1} \times \frac{F}{\binom{K}{t}} + N \times \binom{K-1}{t-1} \times \frac{lF}{r\binom{K}{t}} \\ &= \frac{NtF}{K}\left(1 + \frac{l}{r}\right) = M_2F. \end{aligned} \tag{12}$$

Remark 1. *Note that the cached contents by any set of at most l untrusted caches do not reveal any information about the library files, thanks to the secure regenerating codes [10], i.e., condition (3) is surely satisfied. Also, the encoding scheme is known to all the end users, i.e., from its cached contents, subfiles and random symbols, user k can regenerate the coded symbols $C^j_{n,\mathcal{T}}$, for all n and j as long as $k \in \mathcal{T}$.* ■

B. Delivery Phase

We focus on the worst-case demand, where all the end users request different files. Consider all subsets of the users of size $t+1$. We denote this sets by $\mathcal{H}_i$, $i = 1, ..., \binom{K}{t+1}$. For the untrusted cache j, we consider the sets $\mathcal{H}_i$ such that $\mathcal{H}_i \cap \mathcal{K}_j \neq \phi$. In particular, Γ_j transmits to the users in $\mathcal{H}_i \cap \mathcal{K}_j$, the signal

$$X^{\mathcal{H}_i}_{j,\boldsymbol{d}} = \bigoplus_{\{k : k \in \mathcal{H}_i \cap \mathcal{K}_j\}} C^j_{d_k, \mathcal{H}_i \setminus \{k\}}. \tag{13}$$

In total, the untrusted cache j transmits, the following signal

$$X_{j,\boldsymbol{d}} = \bigcup_{\mathcal{H}_i : \mathcal{H}_i \cap \mathcal{K}_j \neq \phi} \{X^{\mathcal{H}_i}_{j,\boldsymbol{d}}\}. \tag{14}$$

Note that user k can regenerate all coded symbols involved in $X^{\mathcal{H}_i}_{j,\boldsymbol{d}}$ except $C^j_{d_k, \mathcal{H}_i \setminus \{k\}}$. Thus, it obtains $C^j_{d_k, \mathcal{H}_i \setminus \{k\}}$ from the received signal from untrusted cache j. Now, since user k is connected to r different relay nodes, it will obtain $C^j_{d_k, \mathcal{H}_i \setminus \{k\}}$ for all $j \in \mathcal{N}(U_k)$. Thanks to the secure regenerating encoding, it will be able to decode $W_{d_k, \mathcal{H}_i \setminus \{k\}}$. Taking into account all the possible choices of $\mathcal{H}_i$, in addition to the cached subfiles by user k, it can reconstruct its requested file W_{d_k} by the end of the delivery phase.

C. Delivery Load Calculations

Note that the delivery load is a function in the network connectivity parameters, i.e., k_j's. In particular, the untrusted cache j transmits $\left|\left\{ i \in \{1, ..., \binom{K}{t+1}\} : \mathcal{H}_i \cap \mathcal{K}_j \neq \phi \right\}\right|$ subsignals each of length equals to $\frac{F}{(r-l)\binom{K}{t}}$ bits. Thus, we get

$$\begin{aligned} R_jF &= \left|\left\{ i \in \left[\binom{K}{t+1}\right] : \mathcal{H}_i \cap \mathcal{K}_j \neq \phi \right\}\right| \frac{F}{(r-l)\binom{K}{t}} \\ &= \left(\binom{K}{t+1} - \left|\left\{ i \in \left[\binom{K}{t+1}\right] : \mathcal{H}_i \cap \mathcal{K}_j = \phi \right\}\right| \right) \\ &\qquad \times \frac{F}{(r-l)\binom{K}{t}} \\ &= \left(\binom{K}{t+1} - \binom{K-k_j}{t+1} \right) \frac{F}{(r-l)\binom{K}{t}}. \end{aligned} \tag{15}$$

Whenever, $M_2 \geq N$, each of the end users can store the entire library and there is no transmission during the delivery phase, i.e., $R_j = 0$. For any value of M_2, the achievable delivery load is defined as the lower convex envelope of the considered points, which is obtained by memory sharing [1].

In Fig. 2, we compare the total delivery load for different values of l, i.e., number of colluding untrusted caches, considering a network topology where $k_1 = 7$, $k_2 = 6$, $k_3 = 4$, $k_4 = 3$, and $k_5 = k_6 = 2$. At each case, we assume that the total memory at the intermediate layer is just enough to store the database library while maintaining the secrecy requirements, i.e., $M_1 = \frac{N}{r-l}$. The case where $l = 0$, represents the scenario where all caches are assumed to be

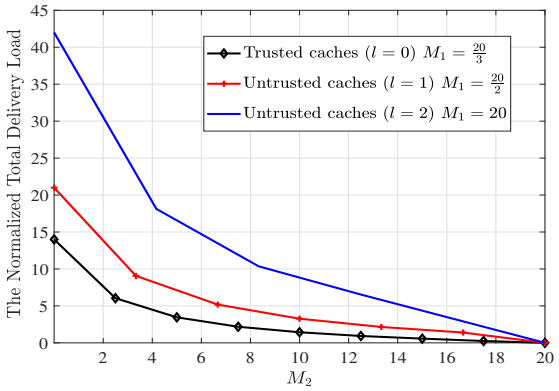

Fig. 2: The total normalized delivery load for $N=20$, $K=8$, $h=6$, $r=3$, $k_1=7$, $k_2=6$, $k_3=4$, $k_4=3$, and $k_5=k_6=2$.

trusted [4]. Clearly, as l increases the delivery load increases, showing the cost of cooperation with untrusted caches.

IV. Achievable Normalized Total Delivery Time

We next calculate the total normalized delivery time of our proposed scheme under the two modes of transmission.

A. Successive Transmission

The achievable normalized total delivery time is given by

$$T_{\text{suc}} = \sum_{j=1}^{h} R_j = \sum_{j=1}^{h} \frac{\binom{K}{t+1}}{(r-l)\binom{K}{t}} - \frac{\binom{K-k_j}{t+1}}{(r-l)\binom{K}{t}}$$
$$= \frac{h(K-t)}{(r-l)(t+1)} - \frac{\sum_{j=1}^{h}\binom{K-k_j}{t+1}}{(r-l)\binom{K}{t}}. \quad (16)$$

Similar to [4], due to the convex nature of the binomial coefficients in (16), we observe that the minimum value for the total delivery time occurs when all end users are connected to the same set of r relays, i.e., $k_j=K$ and $k_i=0$ for all $i\neq j$. In this case, the normalized delivery time is given by

$$T_{\text{suc}}^{\min} = \frac{r(K-t)}{(r-l)(t+1)}. \quad (17)$$

The topologies that maximize the total delivery time are the ones with $\max_j k_j \leq \min_j k_j + 1$, i.e., the connectivity is almost uniform among all the untrusted caches [4].

The average normalized total delivery time can be calculated by averaging the expression in (16) over all possible network topologies [4]. In particular, the total number of possible network topologies is $\binom{h}{r}^K$. Let $N(k_j)$ be the number of possible topologies where the untrusted cache j is connected to k_j end users, i.e., we have $N(k_j)=\binom{K}{k_j}\binom{h-1}{r-1}^{k_j}\binom{h-1}{r}^{K-k_j}$. The average of the normalized total delivery load, over all network topologies, is given by

$$E[T_{\text{suc}}] = \frac{h(K-t)}{(r-l)(t+1)} - \frac{h}{(r-l)\binom{K}{t}}\sum_{k_j=1}^{K} P(K_j=k_j)\binom{K-k_j}{t+1}, \quad (18)$$

where $P(K_j=k_j)=\frac{N(k_j)}{\binom{h}{r}^K}$.

B. Simultaneous Transmission

In this case, the total normalized delivery time is given by

$$T_{\text{sim}} = \max_j \left(\binom{K}{t+1} - \binom{K-k_j}{t+1}\right)\frac{F}{(r-l)\binom{K}{t}}. \quad (19)$$

The topologies that maximize the total delivery time are the ones where all end users are connected to the same set of r untrusted caches. In contrast, the topologies that minimize the total delivery time are the ones where the numbers of end users served by each of the untrusted caches are close as possible to each other [4].

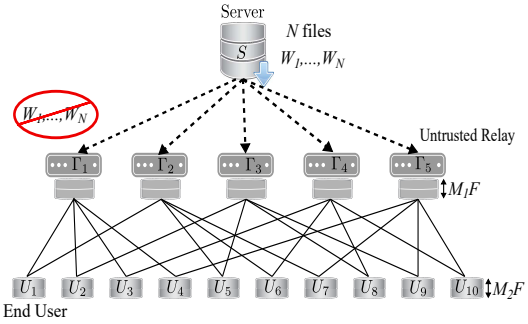

Fig. 3: A combination network with $K=10$, $h=5$ and $r=2$.

V. Extensions and Discussion

A. Combination Networks with Untrusted Relays

Here, we focus on a specific network topology known as *combination networks* [3], with untrusted relay nodes. In such setup, a server is connected to a set of $K=\binom{h}{r}$ end users via h relay nodes such that each user is connected to a distinct set of r relay nodes via unicast links, as shown in Fig. 3. Again, we assume that each relay is equipped with a cache memory of size M_1F bits while each end user is equipped with a cache of size M_2F bits. The relays are untrusted, and any l relays can collude. We consider the case where $(r-l)M_1+M_2\geq N$. In this case, the caches in the network can collectively disengage the server from the delivery phase. Due to the unicast connectivity between the untrusted caches and the end users, i.e., no possibility of a multicast transmission, there is no need to overlap contents of the caches of the end users and the untrusted caches. For any $0\leq M_2\leq N$, we divide each file into two subfiles: W_n^1 of size $\frac{M_2F}{N}$ bits and W_n^2 of size $F-\frac{M_2F}{N}$ bits. W_n^1's will be cached by the end users, i.e., $Z_k=\{W_n^1:\ \forall n\}$. Clearly, this allocation satisfies the memory constraint at the end users. The subfiles W_n^2 will be encoded using (h,r,l) secure regenerating code as explained in Section III. Therefore, we obtain the coded symbols $C_n^1,..,C_n^h$ each of size $\frac{1}{r-l}\left(F-\frac{M_2F}{N}\right)$ bits. Each of the untrusted caches will store one coded symbol from each file, i.e., $V_j=\{C_n^j:\ \forall n\}$. Under the total memory constraint, $(r-l)M_1+M_2\geq N$, the memory capacity constraints are surely satisfied.

During the delivery phase, user k retrieves the subfile $W^1_{d_k}$ from its cache and downloads the coded symbols C^j_n from the untrusted relays in $N(U_k)$. Thus, user k decodes the subfile $W^2_{d_k}$ and hence reconstructs its requested file, successfully. Each untrusted relay is connected to $\binom{h-1}{r-1}$ end users, thus the total normalized delivery load by each untrusted relay can be upper bounded by

$$R_j \leq \frac{Kr}{h(r-l)}\left(1-\frac{M_2}{N}\right). \tag{20}$$

B. Minimum Total Memory

Note that $(r-l)M_1+M_2=N$ represents the total memory in the system is just sufficient to store the database without violating the secrecy requirements [10]. To verify this, assume that user 1 is connected to the first r relays, the cache of user 1 and its connected untrusted caches must store the N files without violating the secrecy constraints, thus we have

$$\begin{aligned} NF &= H(W_1,...,W_N) = H(W_1,...,W_N|V_1,...,V_l) \\ &\quad - H(W_1,...,W_N|V_1,...,V_r,Z_1) && (21)\\ &= I(W_1,...,W_N;V_{l+1},...,V_r,Z_1|V_1,...,V_l) && (22)\\ &\leq H(V_{l+1},...,V_r,Z_1|V_1,...,V_l) && (23)\\ &\leq H(V_{l+1},...,V_r,Z_1) && (24)\\ &\leq M_2F+(r-l)M_1F. && (25) \end{aligned}$$

In this case, the total memory in the system is not enough to create multicast opportunities during the delivery phase, i.e., no overlap between the contents of the untrusted caches and the end users. The allocation scheme is similar to the one described in Section V-A. The untrusted caches transmit unicast signals to each of the connected end users, i.e., we get

$$R_j = \frac{k_j}{r-l}\left(1-\frac{M_2}{N}\right), \tag{26}$$

$$T_{\text{sim}} = \frac{\max_j k_j}{r-l}\left(1-\frac{M_2}{N}\right), \quad T_{\text{suc}} = \frac{rK}{r-l}\left(1-\frac{M_2}{N}\right). \tag{27}$$

C. Size of Untrusted Caches

When $M_1 > \frac{N}{r-l}$, we are able to create some redundancy in the untrusted caches. We consider the cases where $M_1 = \frac{N}{r-z-l}$ for some integer $z \in [r-1]$ and $r-z > l$. The scheme for other values of M_1 can be obtained by memory-sharing technique [1]. For a given M_1, we encode each subfile using an $(h, r-z, l)$ secure regenerating code, i.e., each user is able to reconstruct any requested file by connecting to $r-z$ servers. The secrecy requirements necessitate that $l < r-z$. The placement at the end users is performed as in Section III, while each untrusted cache will store of the resulting coded symbols. During the delivery phase, each user connects to a random set of r untrusted caches. We have a degree of freedom thanks to the additional storage capability at each of the untrusted caches. Each user can retrieve a reconstruct one of the requested subfiles from the signals received from only a subset of $r-z$ untrusted caches. The untrusted cache that will deliver the coded symbols is chosen such that the multicast opportunities across the network are maximized as explained in [4].

D. Secure Delivery as a Byproduct

The proposed scheme, in Section III, ensures that if any external eavesdropper accesses the transmitted signals by any set of at most l untrusted caches during the delivery phase, it cannot gain any information about the database files. More formally, for $\epsilon > 0$, we have

$$I(W_1,...,W_N;\{X_{j,\boldsymbol{d}}\}_{j\in\mathcal{S}}) \leq \epsilon, \quad \mathcal{S}\subset[h], |\mathcal{S}| \leq l. \tag{28}$$

Note that the eavesdropper, here, is assumed to have limited access for the signals transmitted during the delivery phase, i.e., it can access the signals transmitted by at most l untrusted caches, unlike the case in [3], [5], [6], where the eavesdropper overhears all the transmitted signals during the delivery phase.

VI. Conclusions

In this work, we have investigated cache-aided networks, where a layer of h untrusted caches servers a group of cache-equipped end users. In particular, we have required that any collusion formed by at most l untrusted caches must not gain any information about the database files. During the delivery phase, each end user connects randomly to a set of r untrusted caches. By utilizing secure regeneration coding and jointly optimizing both cache placement and delivery phases, we have proposed an achievability scheme that satisfies the users' requests while ensuring the confidentiality of the database files at the layer of untrusted caches. Additionally, we have extended our achievability to combination networks with cache-aided untrusted relays.

Future directions include cooperation with Byzantine caches, i.e., may provide false data, and considering untrusted caches at different layers.

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] A. A. Zewail and A. Yener, "Combination networks with or without secrecy constraints: The impact of caching relays," *IEEE Journal on Selected Areas in Communications*, 2018.

[4] N. Mital, D. Gündüz, and C. Ling, "Coded caching in a multi-server system with random topology," in *Wireless Communications and Networking Conference (WCNC), 2018 IEEE*. IEEE, 2018.

[5] 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.

[6] 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.

[7] ——, "Fundamental limits of secure device-to-device coded caching," in *50th Asilomar Conference on Signals, Systems and Computers*, 2016.

[8] A. A. Zewail and A. Yener, "The wiretap channel with a cache," in *IEEE International Symposium on Information Theory (ISIT)*, 2018.

[9] 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.

[10] K. Rashmi, N. B. Shah, K. Ramchandran, and P. V. Kumar, "Information-theoretically secure erasure codes for distributed storage," *IEEE Trans. Info. Theory*, vol. 64, no. 3, pp. 1621–1646, 2018.