IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 1, 2019, accepted January 11, 2020, date of publication January 17, 2020, date of current version January 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2967427

Weakly Secure Coded Caching Scheme for an Eavesdropper Having Prior Knowledge

NAN WANG[1,3], HAI ZHAO[1], HAIBO JIN[4], AND LONG HAI[2]

[1]Computer Science and Engineering, Northeastern University, Shenyang 110819, China
[2]School of Computer Engineering, Shenzhen Polytechnic, Shenzhen 518055, China
[3]School of Information Engineering, Shenyang University, Shenyang 110044, China
[4]College of Software Engineering, Liaoning Technical University–Huludao, Huludao 125105, China

Corresponding author: Long Hai (hailong@szpt.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 61701314 and Grant 51908366, and in part by the Funding from Scientific and Technical Innovation Council of Shenzhen Government under Grant JCYJ20170818102237376, Grant JCYJ20170818144026871, Grant JCYJ20170818101314276, and Grant JCYJ20170818140234295.

ABSTRACT Coded caching is a promising method for solving caching problems in content-centric wireless networks. To enhance the security of coded caching for practical purposes, this paper investigates a secure coded caching scheme for defending against an eavesdropper who may possess prior knowledge before eavesdropping on content delivery. A novel key-based solution is proposed to minimize the leakage of meaningful information to the eavesdropper during the delivery phase. In the proposed solution, the central server observes the strength of the eavesdropper and introduces several keys to dynamically encrypt the broadcast signals within a weak security constraint. The amount of cache memory occupied by the keys is carefully designed to achieve the highest possible cache gain in terms of the rate of content delivery. The tradeoff between the cache memory usage and the transmission rate with varying eavesdropper strength is characterized. For a given size of the total cache memory, the tradeoff between the amounts of memory used for data and key storage is also presented. We report simulation results to support our analytical findings and show that no keys are needed to achieve weak security against a low-strength eavesdropper. Even for an eavesdropper of high strength, weak security can be introduced at a negligible cost given a large cache size.

INDEX TERMS Network security, communication networks, network coding, cooperative caching.

I. INTRODUCTION

The exponential growth of mobile cellular network traffic has led to an explosion of ''content-centric'' communications such as video streaming and content sharing. To address the extremely large associated traffic demands, one feasible method is to leverage ubiquitous caching in wireless networks during off-peak operation. This wireless caching method has enormous potential to break the current bottleneck for massive content delivery. The general caching problem has been well studied in the literature [1]–[3]. Most previous works have considered a fixed delivery scheme and have optimized the storage phase to achieve gains through local content distribution [4], [5]. Recently, a novel method called coded caching [6]–[8] has been proposed, in which files are broken up during the placement phase and gains from global cache interactions are achieved by index coding technology. Coded caching enables a multiplicative improvement in peak transmission rates (transmissions from the server under high numbers of user requests).

Although coded caching enables a significant gain in terms of the peak rate, the caching of unencrypted data and the network-coded multicast or broadcast nature of the transmission introduce security issues to content delivery. Sengupta *et al.* have proposed an interesting method for solving the problem of insecure coded caching to defend against a simple external adversary who is an eavesdropper without any prior knowledge of the content being transmitted [9]. A similar idea has been applied in the cases of decentralized caching [10] and nonuniform user demands [11]. In practice, however, the eavesdropper may not be a simple adversary without prior knowledge. For example, some users may leak some files or keys to the eavesdropper during the caching process. For the case in which the eavesdropper has prior

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaofan He.

knowledge of the cached files and keys, the question of how to enhance the security of content delivery remains open. On the other hand, the security requirement may not always be strict. For example, consider an eavesdropper with access to one data bit, $b_1 + b_2$. Although the eavesdropper has one bit of information about the source, he is unable to recover any of the source bits if he has neither of the data bits b_1 and b_2. Such coded data may be sufficiently secure for some applications, although they are not information theoretically secure. If the security requirement is weakened to suit practical requirements, the transmission rate loss in unicast and multicast scenarios can be reduced [12]. In view of the above problems, this paper investigates a secure coded caching problem with the following two scenarios: (a) a powerful eavesdropper who may possess prior knowledge of the cached content and who eavesdrops on the content delivery phase to steal meaningful information and (b) a weak security constraint in which multiple messages are combined to limit the eavesdropper's ability to extract any meaningful information about the content being delivered. Compared with Shannon security, weak security incurs a lower encryption cost to satisfy the desired security constraint. Therefore, fewer transmission resources can be used for encryption keys. For details on weakly secure codes, the reader is referred to [13] and [14].

To address the considerations discussed above, we propose a weakly secure coded caching scheme to enhance the security of content delivery against a powerful eavesdropper. The main contributions of this paper are summarized as follows.

- We propose a weakly secure coded caching scheme in which the central server observes the strength of the eavesdropper and selectively generates random keys with which to encrypt the broadcast signals. Under the weak security constraint, the amount of cache memory occupied by the keys is optimized to achieve the highest possible cache gain.
- We analyze the tradeoff between the cache memory usage and the content delivery rate for eavesdroppers of different strengths and present the bounds on the achievable rate for different cache gains.
- For a given total size of the cache memory, we analyze the tradeoff between the amounts of memory used for data and key storage for different parameters of the coded caching model.

The remainder of this paper is organized as follows. Section II presents some related work on secure and insecure coded caching schemes. Section III describes the system model considered in this paper. In section IV, we propose our weakly secure coded caching scheme and discuss the memory-rate and data-key tradeoffs for different parameters of the coded caching model. An analysis of numerical results is presented in section V. Finally, we conclude the paper in section VI.

II. RELATED WORK

Coded caching, as proposed by Maddah-Ali *et al.*, introduces an information-theoretic formulation of the caching problem that utilizes the cumulative size of the network cache memory to jointly design the caching and delivery phases based on a network coding method [6]. An additional gain called global caching gain is achieved; this reduces the number of server transmissions moreso than traditional wireless caching at the peak rate of content requests. This promising idea has motivated a wealth of research efforts. In [15], Maddah-Ali *et al.* further extended their work to the case of decentralized storage and claimed that their decentralized caching scheme achieved a rate close to that of the optimal centralized scheme. Niesen *et al.* investigated coded caching with nonuniform user demands and proposed a coded caching scheme with knowledge of the popularity distribution of the files to be delivered [8]. They also extended the coded caching method to consider delay-sensitive content and proposed three algorithms to achieve coding gains for low-delay delivery [16]. Ji *et al.* introduced a coded caching method for wireless device-to-device (D2D) networks and showed that the spatial reuse gain in a D2D network is order equivalent to the coded multicasting gain in the case of single-base-station transmission [17]. Pedarsani *et al.* considered a caching problem without knowledge of future requests and proposed a corresponding online coded caching scheme [7]. Karamchandani *et al.* extended coded caching to multilayer networks and showed that the layers can simultaneously operate at approximately the minimum rate [18].

The efforts reviewed above represent a promising beginning to research coded caching problems without any security considerations. Although coded caching enables a significant gain in the peak rate, the network-coded broadcast nature of the transmission introduces security issues in content delivery. Recently, Sengupta *et al.* investigated a secure coded caching problem with an external eavesdropper [9]. They assumed that the content was delivered over an insecure link but that the caching phase was handled in a secure manner. Improved methods for the cases of decentralized caching and nonuniform demands were proposed in [10] and [11], respectively. Awan *et al.* introduced this secure coded caching concept into D2D networks and proposed a coding scheme to achieve secure content delivery in the case of an external eavesdropper [19]. Ravindrakumar *et al.* presented research on secretive coded caching, in which a user cannot learn any information about unsolicited files from either the cached content or the server transmissions [20]. In the above line of research, it was assumed that the eavesdropper knew nothing about the transmitted files prior to content delivery. Our work differs from the above research in that it focuses on a powerful eavesdropper who may possess some information about the cached content (data files and keys) and eavesdrops on the content delivery phase to steal meaningful information. In addition, the above works guarantee Shannon security for content delivery. However, the key length needed for Shannon security may be excessively expensive for certain practical applications. Reducing the key length by relaxing the security requirement is reasonable when defending against certain computationally limited eavesdroppers. In light of this

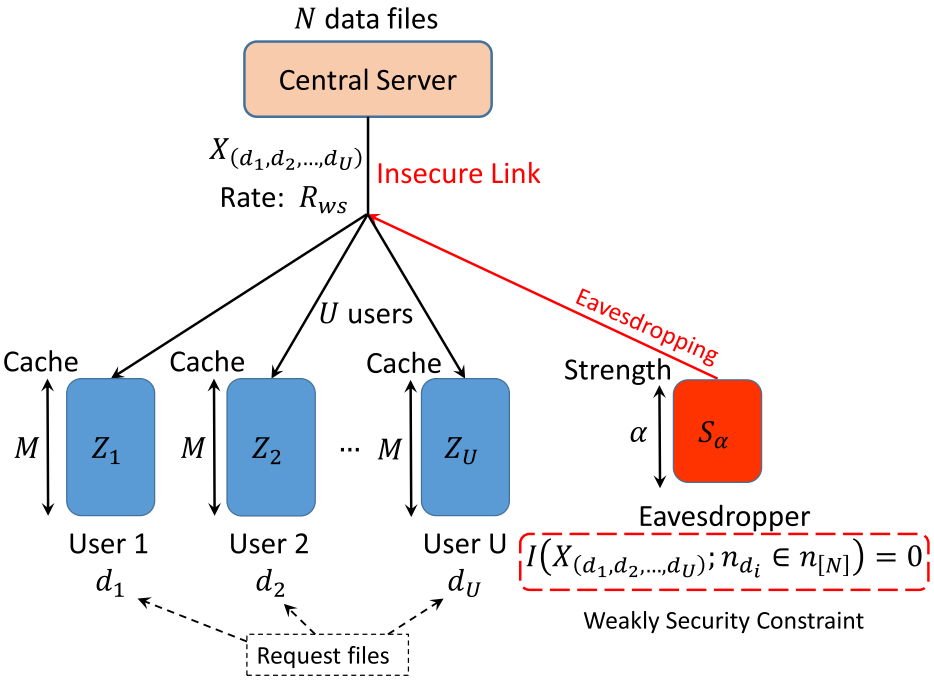


FIGURE 1. System model for weakly secure caching.

consideration, we introduce weak security [13] into the secure coded caching problem. In the case of weak security, data are encrypted with other data or shortened keys. Such a security constraint has been well studied in [21], [22] and [23]. By combining weak security with coded caching, a feasible key solution for defending against an eavesdropper with prior knowledge is proposed in this paper.

III. SYSTEM MODEL

We consider a single-hop content delivery network, as shown in Fig. 1. The system model consists of one central server, one eavesdropper E_α and U users $\{1, 2, \ldots, U\} \triangleq [U]$. All users connect to the central server via an error-free shared link. The central server contains N independent data files $\{n_1, n_2, \ldots, n_N\} \triangleq n_{[N]}$, where $\{1, 2, \ldots, N\} \triangleq [N]$. Each of the data files is uniformly distributed over

$$\left[2^F\right] \triangleq \left\{1, 2, \ldots, 2^F\right\} \tag{1}$$

with a size of F bits for some $F \in \mathbb{N}$. To defend against the eavesdropper, the central server selectively generates several random keys $[K] \in \mathcal{K}$ as the key files such that K is sufficiently large. Each $k \in [K]$ is independent and has a size of $F_k \leq F$. Each user $u \in [U]$ has a cache memory of MF bits in size, where M can be any real number $M \in [0, N]$. The cache memory can be divided into a data memory Z_u^D and a key memory Z_u^K, with sizes of $M_D F$ and $M_K F$, respectively, i.e., $M = M_D + M_K$. The cache gain for all data files is expressed as

$$g = \frac{UM_D}{N}. \tag{2}$$

Similar to the model proposed in [6], the system operates in two distinct phases: a *placement phase* and a *delivery phase*. In the placement phase, the central server stores in the cache of user u some content Z_u, which is a function of $[n_N]$ and $[K]$. In the delivery phase, each user u requests one of the files $n_{d_u} \in [n_N]$ from the central server. The server is informed of these requests and transmits a signal $X_{(d_1,\ldots,d_U)}$ of $R_{ws}F$ bits in size over the shared link, where R_{ws} is some fixed real number. In this paper, we only focus on the static scenario, in which all users (even mobile users) are always within the communication range of the server during the storage and delivery phases. Considering the weak security constraint, some random keys may be selected to encrypt $X_{(d_1,\ldots,d_U)}$ in the delivery phase. Thereupon, keys will occupy $M_K F$ bits of the cache for each user in the placement phase. By utilizing the content Z_u and the received encrypted signal $X_{(d_1,\ldots,d_U)}$, each user u intends to reconstruct the requested file n_{d_u}. The eavesdropper E_α observes the system and eavesdrops on the delivery phase. It is assumed that E_α has the ability to access the placement phase and/or some caches. Note that the eavesdropper is distinct from the users. The eavesdropper never communicates with the server in the system. Some cached files are leaked to E_α before the delivery phase. We assume that data files $L_D \subset [n_N]$ and key files $L_K \in [K]$ are leaked to E_α, where $\alpha = |L_D| + |L_K| = g + s$ is a function of g, with s being an integer number satisfying $s \geq -g$. Here, the notation $||$ represents the cardinality of a set. Let $S_\alpha = L_D + L_K$ denote all the leaked files. We refer to an E_α with knowledge of S_α as an *α-strength eavesdropper*, and we use $\widehat{N}_\alpha \triangleq \left\{n_{[N]} \setminus L_D\right\}$ to denote the files in which E_α is interested. Note that it may be possible to learn the strength of the

eavesdropper from historical statistics. For example, suppose that the central server knows that there is one eavesdropper in the network. With statistical results from previous feedback, one file may be leaked out of every 10 transmitted files on average. Thus, the server will design the placement and delivery phases based on this prior knowledge. Alternatively, the server may set security levels for the transmitted files. For example, some files may be more important than others. The server will need to set a higher security level to transmit these files. In our scheme, the server can assume that there is an eavesdropper of high strength. Therefore, the secure caching scheme is designed to defend against most eavesdroppers. It is possible that only some parts of a file may be leaked to the eavesdropper in our system. We assume that a file is considered to be leaked if any part of that file is leaked. Therefore, knowledge of the detailed content of the leaked files is unknown to the central server; only the total number of such files is known. To avoid E_α obtaining any meaningful information on $\widehat{N}_\alpha$ during the delivery phase, we define a weak security constraint as follows.

Definition 1: We say that the delivered files $X_{(d_1,\ldots,d_U)}$ are weakly secure *given the knowledge S_α about all files in $n_{[N]}$ if, for each file $n_i \in n_{[N]}$, we always have*

$$I\left(n_i; X_{(d_1,\ldots,d_U)}|S_\alpha\right) = 0. \tag{3}$$

The above definition of weak security guarantees that the delivered files are secure for each single file n_i but not that they are secure for any combination of files. Note that weak security is in contrast to traditional Shannon security, which guarantees that the delivered files are secure for all files $n_{[N]}$. The mutual information for the above case with the Shannon security criterion can be expressed as

$$I\left(n_{[N]}; X_{(d_1,\ldots,d_U)}|S_\alpha\right) = 0. \tag{4}$$

Based on its knowledge of the strength of the eavesdropper, the central server controls the placement of data files and keys in the cache of each user and processes the transmission of $X_{(d_1,\ldots,d_U)}$ at a rate of R_{ws} while preventing E_α from obtaining any meaningful information about the data files. Let $\overline{n}_{d_u}$ be the file that is finally recovered by user u. The probability of recovery error is defined as

$$P_e \triangleq \max_{u\in[U]} \Pr\left(\overline{n}_{d_u} \neq n_{d_u}\right). \tag{5}$$

The information leaked to the eavesdropper is defined as

$$L \triangleq \max_{(d_1,\ldots,d_U)\in(n_{[N]})^U} \sum_{n_i\in n_{[N]}} I\left(n_i; X_{(d_1,\ldots,d_U)}|S_\alpha\right). \tag{6}$$

Thus, the memory-rate tradeoff with weak security is defined as follows.

Definition 2: The memory-rate pair (M, R_{ws}) is achievable for weak security if there exists an (M, R_{ws}) weakly secure caching scheme with $P_e \leq \epsilon$ and $L \leq \epsilon$ for any $\epsilon > 0$ and every sufficiently large file size F. Let $\Lambda_{N,U,\alpha}$ denote the region containing all such achievable pairs. The memory-rate tradeoff with weak security is defined as

$$R^*_{ws}(M) \triangleq \inf\{R_{ws} : (M, R_{ws}) \in \Lambda_{N,U,\alpha}\}. \tag{7}$$

IV. CODED CACHING SCHEME WITH WEAKLY SECURE DELIVERY

In this section, we discuss a coded caching scheme that is aimed at achieving weakly secure delivery to defend against eavesdropping by an α-strength eavesdropper. Before going into detail, we briefly review the original coded caching scheme.

A. CODED CACHING SCHEME

We have already stated that the entire coded caching scheme can be divided into two phases: the placement phase and the delivery phase. In the placement phase, given the number of files N, the number of users U and the cache size MF for each user, the central server computes the cache gain $g = UM/N$. All data files are split into equally sized fragments:

$$\left(n_{i,\mathcal{T}}\right) \triangleq \left\{n_{i,\mathcal{T}} : \mathcal{T} \in \{\mathcal{T} \subset [U] : |\mathcal{T}| = g\}\right\}. \tag{8}$$

An equal size for all fragments is required for the xor encoding operation. For each user $u \in [U]$, the file fragments $\left(n_{i,\mathcal{T}}\right)$ are placed into the cache Z_u for all $i \in [N]$ and $u \in \mathcal{T}$. In the delivery phase, according to the request files $\{n_{d_1}, \ldots, n_{d_U}\}$, the central server broadcasts a set of encoded signals $X_{(d_1,\ldots,d_U)} = \oplus_{u\in\mathcal{S}} n_{d_u,\mathcal{S}\setminus\{u\}}$ to the requesting users for all $\mathcal{S} \in \{\mathcal{S} \subseteq [U] : |\mathcal{S}| = g + 1\}$, where $\mathcal{S}$ defines a collection of special subsets in $[U]$ and the notation $\mathcal{S} \in \{\mathcal{S} \subseteq [U] : |\mathcal{S}| = g + 1\}$ represents the collection of all subsets of $[U]$ such that the size of each subset is equal to $g + 1$. Each user u receives the coded signals and utilizes content in the cache Z_u to recover the requested files. Note that we omit the random delay of the wireless broadcast here. Our scheme is not sensitive to these types of delays. Such broadcast delay can also be optimized by wireless scheduling methods [24], [25] and joint decoding delay methods [26], [27]. The coded caching method described above achieves a multiplicative improvement in the savings of the peak rate during the delivery phase [6].

Reviewing this coded caching scheme, we find the interesting result stated in the following lemma with regard to weakly secure delivery.

Lemma 1: If each user in $[U]$ requests only one distinct file each time, the signal $X_{(d_1,\ldots,d_U)}$ broadcast by the central server using the coded caching scheme is inherently weakly secure against an α-strength eavesdropper if $\alpha < g$.

Proof: Showing that the signal $X_{(d_1,\ldots,d_u)}$ is weakly secure is equivalent to showing that for all $n_{d_u} \in n_{[N]}$,

$$I\left(n_{d_u}; X_{(d_1,\ldots,d_u)}|S_\alpha\right) = 0. \tag{9}$$

Let $X^i_{(d_1,\ldots,d_u)} = \oplus_{u\in\mathcal{S}_i} n_{d_u,\mathcal{S}_i\setminus\{u\}}$, with $\mathcal{S}_i \subseteq [U] : |\mathcal{S}_i| = g + 1$, be the ith transmission for delivering $X_{(d_1,\ldots,d_U)}$. According to the coded caching scheme, the encoded content of each coded transmission is different, i.e., $n_x \neq n_k$ if $n_x \in X^i_{(d_1,\ldots,d_u)}$ and $n_k \in X^j_{(d_1,\ldots,d_u)}$ for all $i \neq j$. Hence, the mutual information of each transmission is as follows for all $i \neq j$:

$$I\left(X^i_{(d_1,\ldots,d_U)}, X^j_{(d_1,\ldots,d_U)}\right) = 0. \tag{10}$$

Based on the theory of information entropy, we have

$$\begin{aligned} & I\left(n_{d_u}; X_{(d_1,\ldots,d_u)}|S_\alpha\right) \\ &= \sum_{i=1}^{\binom{U}{g+1}} I\left(n_{d_u}; X^i_{(d_1,\ldots,d_u)}|S_\alpha\right) \\ &= \sum_{i=1}^{\binom{U}{g+1}} \left[H\left(X^i_{(d_1,\ldots,d_u)}|S_\alpha\right) - H\left(X^i_{(d_1,\ldots,d_u)}|S_\alpha|n_{d_u}\right)\right]. \end{aligned} \tag{11}$$

Because $|\mathcal{S}_i| = g+1$, the broadcast signal $X^i_{(d_1,\ldots,d_u)}$ is encoded by $g+1$ file fragments $n_{d_u,\mathcal{S}_i\setminus\{u\}}$. For the eavesdropper, the leaked files S_α include at most α independent files. We adopt the notation $X^i_t \subseteq X^i_{(d_1,\ldots,d_u)}$ with $t \le g+1$. We have

$$H\left(X^i_{(d_1,\ldots,d_u)}|S_\alpha\right) = H\left(X^i_{t_1}\right) \tag{12}$$

and

$$H\left(X^i_{(d_1,\ldots,d_u)}|S_\alpha|n_{d_u}\right) = H\left(X^i_{t_2}\right) \tag{13}$$

with $t_1 \ge g+1-\alpha$ and $t_1-1 \le t_2 \le t_1$. Since $\alpha < g$, we know that $t_1, t_2 > 0$. Thus, $H\left(X^i_{t_1}\right) > 0$, and $H\left(X^i_{t_2}\right) > 0$. Because $X^i_{t_1}$ and $X^i_{t_2}$ have the same bit length $F/\binom{U}{g}$, substituting (12) and (13) into (11) yields

$$\begin{aligned} & I\left(n_{d_u}; X_{(d_1,\ldots,d_u)}|S_\alpha\right) \\ &= \sum_{i=1}^{\binom{U}{g+1}} \left[H\left(X^i_{t_1}\right) - H\left(X^i_{t_2}\right)\right] \\ &= \sum_{i=1}^{\binom{U}{g+1}} \left[\log_2\left(\frac{F}{\binom{U}{g}}\right) - \log_2\left(\frac{F}{\binom{U}{g}}\right)\right] \\ &= 0. \end{aligned} \tag{14}$$

This completes the proof. □

Lemma 1 shows that the original coded caching scheme itself exhibits weak security for a small value of α.

B. WEAKLY SECURE CODED CACHING SCHEME FOR A GENERAL α

In this section, we focus on a general eavesdropper who may have a strength of $\alpha \ge g$. Given the number of files N, the number of users U and the cache size MF, we present a coded caching scheme in Algorithm 1 that incorporates an allocation policy for data files and keys in the placement phase and a transmission policy for requested files in the delivery phase to minimize the meaningful information leaked to the eavesdropper.

In the placement phase, the server computes $g = MU/N$ and selects $M_D = gN/U$ as the size of the data cache memory and $M_K = (1-g/U)\cdot t$, where $t = \max\{\alpha - g + 1, 0\}$, as the size of the key cache memory.[1] If $M - M_D < M_K$, which means that M_K is not sufficiently large for the realization of weak security, then the server reduces the cache gain to $g-1$ and then recomputes $M_D = (g-1)N/U$. Then, N/U of M is reserved for M_K from M_D. Through the execution of this loop, as shown in Lines 2-13, a suitable g is ultimately determined. Then, the server splits each data file $n_i \in n_{[N]}$ into fragments such that $\left(n_{i,\mathcal{T}}\right) \triangleq \left\{n_{i,\mathcal{T}} : \mathcal{T} \in \{\mathcal{T} \subset [U] : |\mathcal{T}| = g\}\right\}$ and stores these fragments in the cache. Because each file is split into $\binom{U}{g}$ fragments, each user should store $\binom{U-1}{g-1}$ fragments of each file. Therefore, the total size of the data files stored in the data cache memory can be computed as

$$M_D = N\frac{\binom{U-1}{g-1}}{\binom{U}{g}} = \frac{gN}{U}. \tag{15}$$

In the delivery phase, each user $u \in [U]$ requests a file $n_{d_u} \in n_{[N]}$. The server is informed of these requests and broadcasts a coded signal

$$X_{(d_1,\ldots,d_U)} = \oplus_{u\in\mathcal{S}} n_{d_u,\mathcal{S}\setminus\{u\}} \tag{16}$$

for all $\mathcal{S} \in \{\mathcal{S} \subseteq [U] : |\mathcal{S}| = g+1\}$. The above process is similar to that in the original coded caching scheme.

Algorithm 1 Weakly Secure Coded Caching

1: **function** CachePlacement(N, U, MF, α, $[n_N]$,$[k_K]$)
2: $g = MU/N$;
3: $t = \max\{\alpha - g + 1, 0\}$;
4: $M_D = gN/U$;
5: $M_K = (1-g/U)\cdot t$;
6: **while** $M - M_D < M_K$ **do**
7: **if** g=0 **then**
8: **return** no feasible key solution.
9: **end if**
10: $g \leftarrow g-1$;
11: $M_D = gN/U$;
12: $t = \max\{\alpha - g + 1, 0\}$;
13: $M_K = (1-g/U)\cdot t$;
14: **end while**
15: **for all** $i \in [N]$ **do**
16: Split data file n_i into equally sized fragments $\left(n_{i,\mathcal{T}}\right) \triangleq \left\{n_{i,\mathcal{T}} : \mathcal{T} \in \{\mathcal{T} \subset [U] : |\mathcal{T}| = g\}\right\}$;
17: **end for**
18: **for all** integer $j \in [1, t]$ **do**
19: Generate random keys $\left(K^j_{\mathcal{T}_k}\right) \triangleq \left\{K^j_{\mathcal{T}_k} : \mathcal{T}_k \in \{\mathcal{T}_k \subseteq [U] : |\mathcal{T}_k| = g+1\}\right\}$;
20: **end for**
21: **for all** $u \in [U]$ **do**
22: $Z^D_u \leftarrow n_{i,\mathcal{T}} : \forall i \in [N]$ if $u \in \mathcal{T}$;
23: $Z^K_u \leftarrow K^j_{\mathcal{T}_k} : \forall j \in [1, t]$ if $u \in \mathcal{T}_k$;
24: **end for**
25: **end function**
26: **function** CodedDelivery($n_{d_1}, \ldots, n_{d_U}$)
27: **for all** $\mathcal{S} \in \{\mathcal{S} \subseteq [U] : |\mathcal{S}| = g+1\}$ **do**
28: $X_{(d_1,\ldots,d_U)} \leftarrow \oplus_{j\in[1,t]} K^j_{\mathcal{S}} \oplus_{u\in\mathcal{S}} n_{d_u,\mathcal{S}\setminus\{u\}}$;
29: **end for**
30: **end function**

[1] In Line 18 of Algorithm 1, we assume that [1, 0] is an empty set for the integer set.

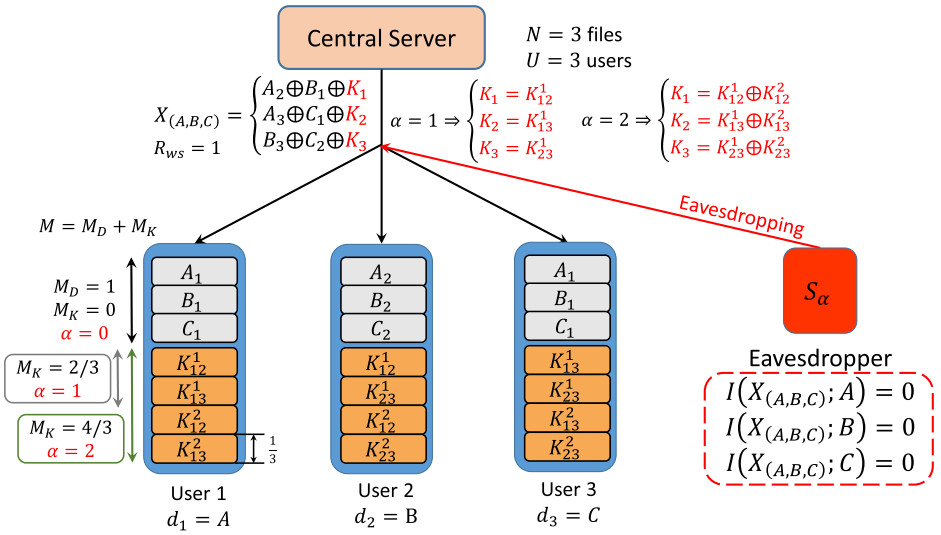


FIGURE 2. Weakly secure coded caching for $N = U = 3$ and $\alpha = \{0, 1, 2\}$.

However, because of the weak security constraint, the length of the coded signal must be at least two fragments greater than the strength α of the eavesdropper. For example, a broadcast signal $A_1 \oplus B_1 \oplus C_1 \oplus D_1$ is weakly secure for $\alpha = 2$ but is not weakly secure for $\alpha = 3$ because an eavesdropper who knows any two of $\{A_1, B_1, C_1, D_1\}$ cannot recover any meaningful signals, whereas the eavesdropper can recover one meaningful signal with a high probability if he knows three of $\{A_1, B_1, C_1, D_1\}$. The server adds t groups of random keys into the signal as follows:

$$X_{(d_1,\ldots,d_U)}|_{ws} = \oplus_{j\in[1,t]} K_{\mathcal{S}}^{j} \oplus_{u\in\mathcal{S}} n_{d_u,\mathcal{S}\setminus\{u\}} \tag{17}$$

where $t = \max\{\alpha - g + 1, 0\}$. Note that the keys $K_{\mathcal{S}}^{j}$ need to be cached during the placement phase. All the keys may also be leaked to the eavesdropper under the weak security constraint. The length of the broadcast signal generated through this process will be at least $\alpha + 2$, thereby guaranteeing that the weak security constraint is satisfied.

C. CACHE MEMORY RESULTS FOR WEAK SECURITY

To achieve this key solution, some keys $K_{\mathcal{S}}^{j}$ must be stored during the placement phase. We use the following example to intuitively illustrate the key allocation performed in Algorithm 1.

Example 1: Consider a case of $N = 3$ files and $U = 3$ users, as shown in Fig. 2. If we fix the cache gain at $g = 1$, then according to (2), the data memory will occupy F bits in the cache. Let the three files be $n_1 = A$, $n_2 = B$ and $n_3 = C$. Each file is split into equally sized fragments, $A = (A_1, A_2, A_3)$, $B = (B_1, B_2, B_3)$ and $C = (C_1, C_2, C_3)$, where all the fragments have an equal size of $F/3$ bits. In the placement phase, the server fills the data memories in the caches of all users with $Z_1^D = (A_1, B_1, C_1)$, $Z_2^D = (A_2, B_2, C_2)$ and $Z_3^D = (A_3, B_3, C_3)$. Suppose that user 1 requests A, user 2 requests B and user 3 requests C. The coded signals $A_2 \oplus B_1$, $A_3 \oplus C_1$ and $B_3 \oplus C_2$ are broadcast if there is no eavesdropping. If there is one eavesdropper of strength $\alpha = 0$, then no part of the file fragments can be recovered because the eavesdropper knows nothing about the files. Thus, the coded signals are weakly secure, and $M = M_D = 1$ because no key is needed in this case. If we suppose that the strength of the eavesdropper is $\alpha = 1$, then at least one key is needed for each coded signal; i.e., $A_2 \oplus B_1 \oplus K_1$ is weakly secure against an eavesdropper who knows any one of A_2, B_1 and K_1. Therefore, the total number of keys is three, i.e., $A_2 \oplus B_1 \oplus K_1$, $A_3 \oplus C_1 \oplus K_2$ and $B_3 \oplus C_2 \oplus K_3$. Because each pair of users shares one key, the following are stored in the key memories of the users: $Z_1^K = (K_1, K_2)$, $Z_2^K = (K_1, K_3)$ and $Z_3^K = (K_2, K_3)$. Because the size of K_1 is equal to the size of the coded signal, which is $F/3$ bits, we have $M = M_D + M_K = 1 + 2/3 = 5/3$. If $\alpha = 2$, then the number of blocks in each coded signal should be increased to at least 4, i.e., $A_2 \oplus B_1 \oplus K_1^1 \oplus K_1^2$. In this case, two groups of keys are needed. Thus, the following are stored in the key memories of the users: $Z_1^K = (K_1^1, K_1^2, K_2^1, K_2^2)$, $Z_2^K = (K_1^1, K_1^2, K_3^1, K_3^2)$ and $Z_3^K = (K_2^1, K_2^2, K_3^1, K_3^2)$. Because the size of each key remains $F/3$ bits, the total cache size is $M = 1 + 4/3 = 7/3$. If $\alpha = 3$, then the number of blocks encoded in the signal should be 5, and we need at least 3 groups of keys. Accordingly, the cache size is $M = 1 + 6/3 = 3$, which is equal to the number of files. In such a case, we can use the cache to store all data files to achieve perfect security instead of storing keys. On the other hand, $\alpha = 3$ also means that the eavesdropper knows all the files of $[N]$, and hence, there is no secure scheme in this case.

Regarding the reservation of space for key storage, one interesting question concerns the size of the data cache memory that remains. As shown in Lines 2 to 13 of Algorithm 1, a dynamic policy is used to compute the size of the data cache. Given this dynamic policy, we use the following lemmas to describe the size of the data cache.

Theorem 1: For a caching scenario with N files, K users, a cache size of $MF = M_D F + M_K F$ for each user and a α-strength eavesdropper, if $\alpha < g$, where $g = KM/N$, then Algorithm 1 is weakly secure for $M_D \leq M$.

Proof: This theorem follows immediately from Lemma 1. □

This theorem follows from Lemma 1 because the coded caching scheme itself possesses weak security. Hence, the full gain of coded caching is achieved for $\alpha < g$ because no keys are needed in this case. For $\alpha \geq g$, however, determining appropriate sizes for M_D and M_K is a complicated problem. For example, consider the case of $N = U = 20$ and $M = 15$. Evidently, if $\alpha < 15$, which means that $\alpha < g = MU/N = 15$, then the case of $M_D = M = 15$ is weakly secure according to Theorem 1. If $\alpha = 15$, at least one key needs to be encoded into each of the broadcast signals. Suppose that the entire cache memory is filled with data such that $M_D = UM/N = 15$. To reserve space for key memory in the cache, the size of the data memory should be reduced. Let us adopt an adjustable parameter $t = g - 1$ such that the new size of the data memory is equal to $M'_D = tN/U = 14$. Now, the size of the memory space reserved for key storage is $M_r = M_D - M'_D = 1$. Without keys, the current coded signal contains $t + 1 = 15$ data blocks. To achieve weak security, at least $15+2-15 = 2$ keys should be added into the coded signal. Then, the amount of key memory occupied in the cache will be at least $2\,(1 - t/U) = \frac{3}{5}$. Because $\frac{3}{5} < M_r$, a data memory size of $M_D = 14$ is achievable for $\alpha = 15$ in this case. Similarly, we have $M_D = 13$ for $\alpha = 16$, $M_D = 12$ for $\alpha = 17$, and $M_D = 10$ for $\alpha = 18$. For all $\alpha \geq 19$, there is no feasible key solution because the cache size M is not sufficiently large. Based on the above discussion, the size of M_D can be derived as stipulated by the following theorem.

Theorem 2: For a weakly secure caching scenario with N files, K users, a cache size of $MF = M_D F + M_K F$ for each user and a α-strength eavesdropper such that $g \leq \alpha \leq M - 1$, Algorithm 1 is weakly secure for $M_D \leq \frac{N}{2U}\left(m + \sqrt{(m-2U)^2 - 4U(N-M)}\right)$ with $m = \alpha + 1 + U - N$.

Proof: According to the key solution shown in (17), we know that the size of each key is equal to $F/\binom{U}{g}$. Because the total number of transmissions is $\binom{U}{g+1}$ (the same as the number of elements in $\mathcal{S}$), to ensure that every g user shares t keys, the number of keys cached by each user must be $\binom{U-1}{g}$. Let $t = \alpha - g + 1$ for $g = KM_D/N$. Hence, the total memory size reserved for key files in the cache of each user can be computed as

$$M_K = t\frac{\binom{U-1}{g}F}{\binom{U}{g}F} = t\left(1 - \frac{g}{U}\right). \tag{18}$$

Substituting (18) and (2) into $M \geq M_D + M_K$, we obtain

$$\begin{aligned} M_D &\leq M - M_K \\ &\leq M - t\left(1 - \frac{\frac{UM_D}{N}}{U}\right) \\ &\leq \frac{N}{N-t}(M - t). \end{aligned} \tag{19}$$

Because t is a function of M_D, substituting $t = \alpha - KM_D/N + 1$ into (19) and rearranging the terms finally yields

$$\frac{U}{N}M_D^2 + (N - U - \alpha - 1)M_D + N(\alpha - M - 1) \leq 0. \tag{20}$$

Solving this inequality and taking the maximal solution yields

$$M_D \leq \frac{N}{2U}\left(m + \sqrt{(m-2U)^2 - 4U(N-M)}\right), \tag{21}$$

where $m = \alpha + 1 + U - N$. □

According to (2), Theorem 2 results in the following cache gain region for the given N, U, M and α:

$$g \leq \max\left\{\frac{1}{2}\left(m + \sqrt{(m-2U)^2 - 4U(N-M)}\right), \frac{UM}{N}\right\}, \tag{22}$$

where $m = \alpha + 1 + U - N$. This result shows that the gain that can be achieved through caching is limited by the strength of the eavesdropper.

D. MEMORY-RATE TRADEOFF FOR WEAKLY SECURE CODED CACHING

Reviewing Definition 2, we can derive an achievable rate that bounds the optimal memory-rate tradeoff $R^*_{ws}(M)$ from above within the weak security constraint.

Theorem 3: For N files, one α-strength eavesdropper and U users, each with a cache size of $M \in \frac{N-t}{U} \cdot g + t$, for $t = \max\{\alpha - g + 1, 0\}$ and $g \in \{0, 1, 2, \ldots, U\}$,

$$R^*_{ws} \leq R^C_{ws}(M) \triangleq U \cdot \left(1 - \frac{M-t}{N-t}\right)\left\{\frac{1}{1 + U\frac{M-t}{N-t}}\right\} \tag{23}$$

is achievable. For any $\alpha + 1 \leq M \leq N$ and $\alpha < U$, the lower convex envelope of these points is achievable.

Proof: Because $M \in \frac{N-t}{U} \cdot g + t$, $g \in \{0, 1, 2, \ldots, U\}$ and $t = \max\{\alpha - g + 1, 0\}$, we have $\alpha + 1 \leq M \leq N$. If g is treated as a variable, the cache memory size can be parameterized as follows:

$$M = \frac{N-t}{U}g + t = \frac{N}{U}g + \left(1 - \frac{g}{U}\right)t. \tag{24}$$

According to (2), the sizes of the data and key cache memories can be expressed as

$$M_D = \frac{N}{K}g; \quad M_K = M - M_D = t\left(1 - \frac{g}{U}\right). \tag{25}$$

If $M = \alpha + 1$, which means that $g = 0$, then we need at least $\alpha + 1$ keys to achieve weak security. The rate satisfies $R^*_{ws}(\alpha + 1) \leq U$. If $M = N$, all data files can be stored in the cache. Then, the rate is $R^*_{ws}(N) = 0$. Therefore, in the following proof, we consider the case of $\alpha + 1 < M < N$ such that $g \in \{1, 2, \ldots, U - 1\}$.

In the placement phase, each data file $n_i \in n_{[N]}$ is split into equally sized fragments: $(n_{i,\mathcal{T}}) \triangleq \left\{n_{i,\mathcal{T}} : \mathcal{T} \in \{\mathcal{T} \subset [U] : |\mathcal{T}| = g\}\right\}$. The total number of fragments of each file is $\binom{U}{g}$. The size of each fragment is thus $F/\binom{U}{g}$ bits. For each i, the fragment $n_{i,\mathcal{T}}$ is stored in the cache

Z_D^u of user u if $u \in \mathcal{T}$. Because $|\mathcal{T}| = g$, user u shares a fragment of a given file n_i with $g-1$ of the $U-1$ other users. Thus, each user caches $N\binom{U-1}{g-1}$ fragments. As shown in Algorithm 1, the total number of transmissions in the delivery phase is $\binom{U}{g+1}$. The number of keys needed to achieve weak security is at least $t\binom{U}{g+1}$, where $t = \max\{\alpha - g + 1, 0\}$. For user $u \in \mathcal{T}_k : |\mathcal{T}_k| = g+1$, a key $K_{\mathcal{T}_k}$ is shared with g out of $K-1$ other users. Thus, there are $t\binom{K-1}{g}$ keys stored in the cache of each user. Note that each key file should be of the same size as the data file fragments, namely, $F/\binom{U}{g}$ bits. Given all these considerations, the number of bits required in the cache of each user is

$$\begin{aligned} & N\binom{N-1}{g-1}\cdot\frac{F}{\binom{U}{g}} + t\cdot\binom{U-1}{g}\cdot\frac{F}{\binom{U}{g}} \\ &= \frac{FNg}{U} + Ft\left(1-\frac{g}{U}\right) \\ &= F\left[\frac{Ng}{U} + t\left(1-\frac{g}{K}\right)\right] \\ &= FM. \end{aligned} \tag{26}$$

In the delivery phase, each user $u \in [U]$ requests one file $n_{d_u} \in n_{[N]}$. Let $\mathcal{S} \subseteq [U]$ be a subset of $|\mathcal{S}| = g+1$ users. Following the placement phase, every g user in $\mathcal{S}$ shares a fragment of a given file, which is requested by the $(g+1)$th user in their caches. Given a user $u \in \mathcal{S}$, the fragment $n_{d_u,\mathcal{S}\setminus\{u\}}$ is requested by u but is known by all users in $\mathcal{S}\setminus\{u\}$ since $|\mathcal{S}\setminus\{u\}| = t$. Therefore, the server broadcasts the coded signal $X_{(d_1,\ldots,d_U)}|_{ws} = \oplus_{j\in[1,t]} K_{\mathcal{S}}^j \oplus_{u\in\mathcal{S}} n_{d_u,\mathcal{S}\setminus\{u\}}$ for each subset $\mathcal{S} \subseteq [U]$. The number of transmissions is $\binom{U}{g+1}$. Each transmission is associated with t keys, i.e., there are $\binom{U}{g+1}$ keys in this caching system. The total number of bits sent over the shared link is

$$\begin{aligned} R_{ws}^C(M)F &= \binom{U}{g+1}\cdot\frac{F}{\binom{U}{g}} \\ &= \frac{U-g}{g+1}\cdot F. \end{aligned} \tag{27}$$

Substituting $g = U(M-t)/(N-t)$ into (27) and rearranging the terms, we obtain

$$R_{ws}^*(M) \le R_{ws}^C(M) = \frac{U\left(1-\frac{M-t}{N-t}\right)}{1+\frac{U(M-t)}{N-t}}. \tag{28}$$

Next, we prove that the delivery phase is weakly secure against a α-strength eavesdropper, i.e.,

$$I\left(n_{d_u}; X_{(d_1,\ldots,d_u)}|S_\alpha\right) = 0. \tag{29}$$

Let $X^i_{(d_1,\ldots,d_u)} = \oplus_{j\in[1,t]} K^j_{\mathcal{S}_i} \oplus_{u\in\mathcal{S}_i} n_{d_u,\mathcal{S}_i\setminus\{u\}}$ be the ith transmission for delivering $X_{(d_1,\ldots,d_u)}$. As observed from the proof of Lemma 1, the data part $\oplus_{u\in\mathcal{S}_i} n_{d_u,\mathcal{S}_i\setminus\{u\}}$ is distinct for all $i \in \left[\binom{U}{g+1}\right]$. Because we associate t distinct keys with each transmission, $X^i_{(d_1,\ldots,d_u)}$ is also distinct for all $i \in \left[\binom{U}{g+1}\right]$. Thus, we have

$$I\left(X^i_{(d_1,\ldots,d_u)}, X^j_{(d_1,\ldots,d_u)}\right) = 0. \tag{30}$$

Then, from the theory of information entropy, it follows that

$$\begin{aligned} & I\left(n_{d_u}; X_{(d_1,\ldots,d_u)}|S_\alpha\right) \\ &= \sum_{i=1}^{\binom{U}{g+1}} I\left(n_{d_u}; X^i_{(d_1,\ldots,d_u)}|S_\alpha\right) \\ &= \sum_{i=1}^{\binom{U}{g+1}} \left[H\left(X^i_{(d_1,\ldots,d_u)}|S_\alpha\right) - H\left(X^i_{(d_1,\ldots,d_u)}|S_\alpha|n_{d_u}\right)\right]. \end{aligned} \tag{31}$$

Because $|\mathcal{S}_i| = t+g+1$, each coded signal $X^i_{(d_1,\ldots,d_u)} \in X_{(d_1,\ldots,d_u)}$ is encoded into $t+g+1$ subfiles (containing data and keys). Let X^i_{t+g+1} represent a signal $X^i_{(d_1,\ldots,d_u)}$ by the number of subfiles it contains. The knowledge possessed by the eavesdropper, S_α, consists of $\max\{t-1,0\}$ such subfiles. Therefore, we have

$$X^i_{(d_1,\ldots,d_u)} - S_\alpha \ge X^i_2. \tag{32}$$

Considering that for each $u \in [U]$, $X^i_{(d_1,\ldots,d_U)}$ contains one fragment of n_{d_u} at most; we therefore know that

$$X^i_{(d_1,\ldots,d_u)} - S_\alpha - n_{d_u} \ge X^i_1. \tag{33}$$

Thus, the entropies satisfy $H\left(X^i_{(d_1,\ldots,d_u)}|S_\alpha\right) \ge H(X^i_2) > 0$ and $H\left(X^i_{(d_1,\ldots,d_u)}|S_\alpha|n_{d_u}\right) \ge H(X^i_1) > 0$. Because all coded signals have the same bit length $F/\binom{U}{g}$, we finally have

$$\begin{aligned} & I\left(n_{d_u}; X_{(d_1,\ldots,d_u)}|S_\alpha\right) \\ &= \sum_{i=1}^{\binom{U}{g+1}} \left[H\left(X^i_{(d_1,\ldots,d_u)}|S_\alpha\right) - H\left(X^i_{(d_1,\ldots,d_u)}|S_\alpha|n_{d_u}\right)\right] \\ &= \sum_{i=1}^{\binom{U}{g+1}} \left[\log_2\left(\frac{F}{\binom{U}{g}}\right) - \log_2\left(\frac{F}{\binom{U}{g}}\right)\right] \\ &= 0, \end{aligned} \tag{34}$$

which proves that the rate $R_{ws}^C(M)$ is achievable for weak security. Thus, the proof of Theorem 3 is complete. □

To intuitively illustrate the result of Theorem 3, we review Example 1 and discuss the memory-rate tradeoff pairs. Because each file is split into three equally sized fragments, the size of each fragment is $F/3$ bits. This indicates that the size of each coded signal is $F/3$ bits. Therefore, the rate of transmission on the shared link is $R_{ws}^C = (3\times F/3)/F = 1$. If $\alpha = 0$, we know that the required cache size is $M = 1$. The memory-rate pair is thus $(M, R_{ws}^C) = (1,1)$. If $\alpha = 1$, the required cache size is $M = 5/3$. The memory-rate pair is thus $(M, R_{ws}^C) = (5/3, 1)$. If $\alpha = 2$, the required cache size is $M = 7/3$, and the memory-rate pair is thus $(M, R_{ws}^C) = (7/3, 1)$. If $\alpha \ge 3$, all data can be stored in the cache. Hence, the memory-rate pair is $(M, R_{ws}^C) = (3, 0)$.

V. NUMERICAL RESULTS

In the following, we present a series of numerical results and evaluate the performance of the proposed scheme in terms

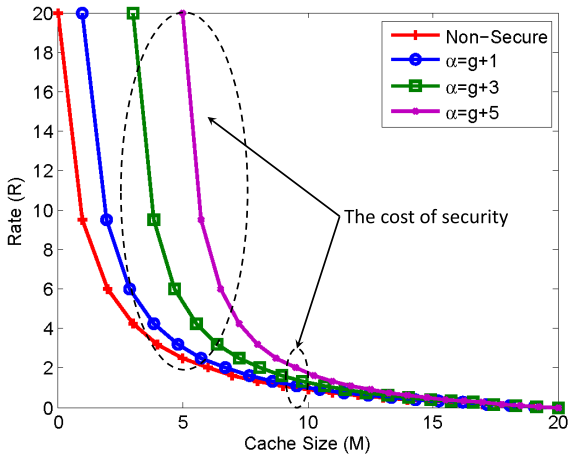


FIGURE 3. Delivery rates with varying cache memory sizes.

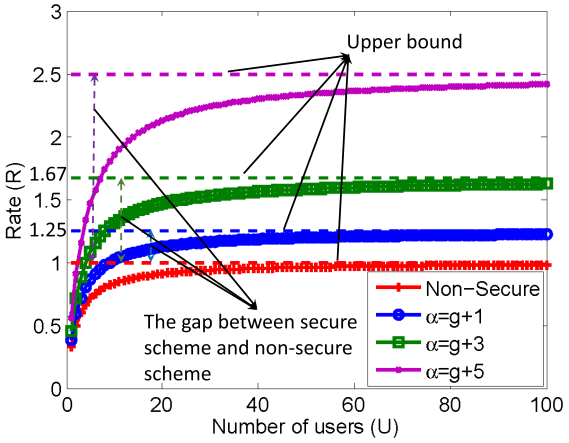


FIGURE 4. Delivery rates with varying numbers of users.

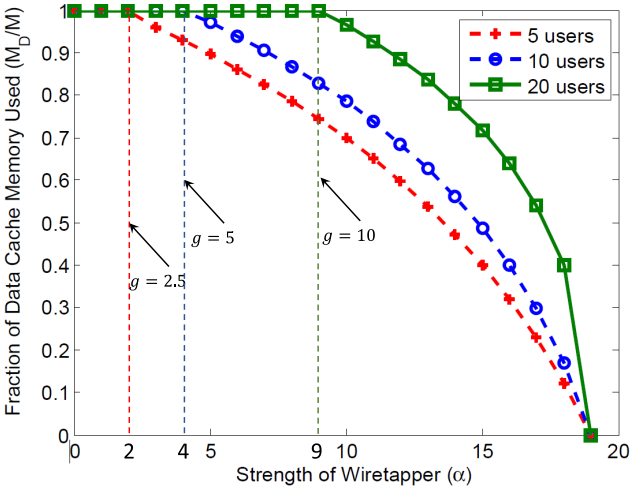


FIGURE 5. Fraction of the cache memory used for data storage as the strength of the eavesdropper varies.

of the delivery rate and cache memory usage. The simulated model is as illustrated in Fig. 1, containing one central server, one α-strength eavesdropper and a number of users, each of whom has a cache of a certain size. All transmissions simulated in this model are transmitted over error-free broadcast links. Both the delivery rates and the cache memory sizes are divided by F to obtain their final values in the simulation results.

A. DELIVERY RATE ON THE SHARED LINK

In this section, we present the delivery rates on the shared link for our proposed scheme in different scenarios. We first consider a case with $N = 20$ files, $U = 20$ users and $s = \alpha - g = \{1, 3, 5\}$. The observed rates on the shared link with varying cache sizes are shown in Fig. 3. As shown, when the cache is small, there is a gap between the weakly secure rates in the cases of large and small α, i.e., the rate for $\alpha - g = 5$ is approximately twice that for $\alpha - g = 1$. The rate achieved with the insecure coded caching scheme serves as a lower bound on that achieved with the weakly secure coded caching scheme because the security requirement incurs some cost for the addition of keys to the signal. As the cache size increases, the rate in the weakly secure case becomes asymptotically equal to that in the insecure case. These results indicate that weak security against an α-strength eavesdropper can be achieved at an *almost negligible cost* for a sufficiently large cache size.

Next, we consider a case with $N = 20$ files, a cache size of $M = 10$ and $s = \alpha - g = \{0, 1, 3, 5\}$. The results for the observed rates on the shared link in the weakly secure case with varying numbers of users are shown in Fig. 4. The rate asymptotically increases as the number of users increases. The gap between the rates for $\alpha - g = 5$ and $\alpha - g = 1$ shows that a cost is incurred in terms of the rate to defend against a high-strength eavesdropper. The rate achieved with insecure-coded caching again serves as a lower bound on the rate of delivery. However, all these results show that the increase in the rate with an increasing number of users is bounded. This indicates that an acceptable rate is achievable for an arbitrary number of users.

B. DATA AND KEY MEMORY USAGE IN THE CACHE

In this section, the amounts of memory used for data and key storage in different scenarios are simulated. We first consider a case with the number of files fixed at $N = 40$ and the size of each cache being $M = 20$. The number of users is limited to $\{5, 10, 20\}$. The fraction of the cache memory used for data storage as a function of the strength α of the eavesdropper is shown in Fig. 5. We can see that the data memory usage goes to 0 as the strength of the eavesdropper increases. However, within a certain range of strength, i.e., $0 \leq \alpha \leq g + 1$, the whole cache memory is filled with data files. This finding shows that our proposed scheme fully utilizes the inherent weak security of coded caching. Even in the range of $g \leq \alpha < 20$, the memory space used for data in the cache forms a convex curve. These simulation results indicate that the cost of security is negative when the strength of the eavesdropper is low.

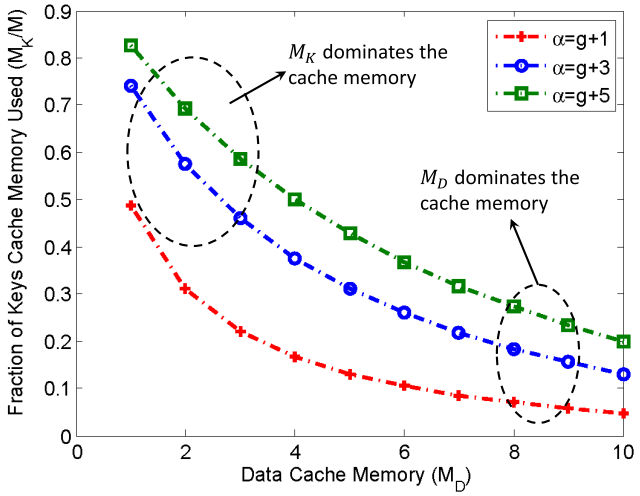


FIGURE 6. Fraction of the cache memory used for key storage as the amount of data cache memory used varies.

Next, we observe the amounts of key cache memory used with different levels of data cache memory usage. We consider the case of $N = 20$ files, $U = 20$ users and an eavesdropper with a strength of $\alpha = \{g+1, g+3, g+5\}$. The simulation results are shown in Fig. 6. The purpose of this simulation setup is to illustrate the tradeoff between data and key memory usage in the cache. As shown, the key memory size decreases as the data memory size increases. This is because an increase in the data memory size causes the cache gain g to increase since $g = M_D U/N$. We also find that the ratio M_K/M decreases dramatically at small α. This shows that the user cache is predominantly used for data storage when the strength of the eavesdropper is low. The above results demonstrate that the key cost is negative for both a low eavesdropper strength and a large amount of cached data.

VI. CONCLUSION

In this paper, we investigate the problem of weakly secure caching in the presence of a powerful eavesdropper who knows some of the cached content before the delivery phase. We show that some security demands can be satisfied by the original coded caching scheme itself without any keys if the strength of the eavesdropper is not high. For the case of a high-strength eavesdropper, we propose a key-based weakly secure caching scheme in which multiple random keys are introduced into the caching system to achieve weak security. The amount of key memory used in the cache is optimized to achieve the highest possible cache gain in terms of the rate of content delivery. The memory-rate tradeoff is presented for eavesdroppers with different strengths. Our analysis shows that the key cost is negative for a large cache memory size and a large number of users. We also reveal the tradeoff between the amounts of memory used for data and key storage for a given cache size. Finally, we evaluate the performance of our scheme through numerical simulations. In our future work, we will extend the current work by considering nonuniform file popularity and multiple eavesdroppers with noisy links.

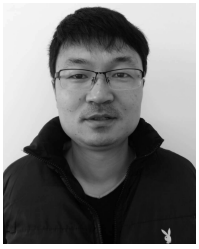
REFERENCES

[1] J. Kwak, Y. Kim, L. B. Le, and S. Chong, "Hybrid content caching in 5G wireless networks: Cloud versus edge caching," *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 3030–3045, May 2018.
[2] F. Xu, M. Tao, and K. Liu, "Fundamental tradeoff between storage and latency in cache-aided wireless interference networks," *IEEE Trans. Inf. Theory*, vol. 63, no. 11, pp. 7464–7491, Nov. 2017.
[3] J. Song and W. Choi, "Minimum cache size and backhaul capacity for cache-enabled small cell networks," *IEEE Wireless Commun. Lett.*, vol. 7, no. 4, pp. 490–493, Aug. 2018.
[4] Z. Chen, N. Pappas, and M. Kountouris, "Probabilistic caching in wireless D2D networks: Cache hit optimal versus throughput optimal," *IEEE Commun. Lett.*, vol. 21, no. 3, pp. 584–587, Mar. 2017.
[5] M. Mangili, F. Martignon, S. Paris, and A. Capone, "Bandwidth and cache leasing in wireless information centric networks: A game theoretic study," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 679–695, Jan. 2017.
[6] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.
[7] R. Pedarsani, M. A. Maddah-Ali, and U. Niesen, "Online coded caching," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 836–845, Apr. 2016.
[8] U. Niesen and M. A. Maddah-Ali, "Coded caching with nonuniform demands," *IEEE Trans. Inf. Theory*, vol. 63, no. 2, pp. 1146–1158, Feb. 2017.
[9] A. Sengupta, R. Tandon, and T. C. Clancy, "Fundamental limits of caching with secure delivery," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 355–370, Feb. 2015.
[10] A. Sengupta, R. Tandon, and T. C. Clancy, "Decentralized caching with secure delivery," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2014, pp. 41–45.
[11] A. Sengupta, R. Tandon, and T. C. Clancy, "Secure caching with non-uniform demands," in *Proc. 4th Int. Conf. Wireless Commun., Veh. Technol., Inf. Theory Aerosp. Electron. Syst. (VITAE)*, May 2014, pp. 1–5.
[12] K. Jain, "Security based on network topology against the wiretapping attack," *IEEE Wireless Commun.*, vol. 11, no. 1, pp. 68–71, Feb. 2004.
[13] K. Bhattad and K. R. Narayanan, "Weakly secure network coding," in *Proc. Winmee Rawnet Netcod Workshops Riva Del Garda*, Apr. 2005, pp. 281–285.
[14] S. H. Dau, V. Skachek, and Y. M. Chee, "On the security of index coding with side information," *IEEE Trans. Inf. Theory*, vol. 58, no. 6, pp. 3975–3988, Jun. 2012.
[15] M. A. Maddah-Ali and U. Niesen, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Trans. Netw.*, vol. 23, no. 4, pp. 1029–1040, Aug. 2015.
[16] U. Niesen and M. A. Maddah-Ali, "Coded caching for delay-sensitive content," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 5559–5564.
[17] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless D2D networks," *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 849–869, Feb. 2016.
[18] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. N. Diggavi, "Hierarchical coded caching," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 3212–3229, Jun. 2016.
[19] Z. H. Awan and R. Mathar, "Bounds on caching D2D networks with secure delivery," in *Proc. 15th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Aug. 2018, pp. 1–5.
[20] V. Ravindrakumar, P. Panda, N. Karamchandani, and V. M. Prabhakaran, "Private coded caching," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 3, pp. 685–694, Mar. 2018.
[21] M. Yan and A. Sprintson, "Weakly secure network coding for wireless cooperative data exchange," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2011, pp. 1–5.
[22] S. H. Dau, V. Skachek, and Y. M. Chee, "On secure index coding with side information," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2011, pp. 983–987.
[23] S. H. Dau, W. Song, and C. Yuen, "Weakly secure MDS codes for simple multiple access networks," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2015, pp. 1941–1945.
[24] Z. Zhang, G. Mao, and B. D. O. Anderson, "Energy-efficient broadcast in mobile networks subject to channel randomness," *IEEE Trans. Wireless Commun.*, vol. 14, no. 6, pp. 2929–2941, Jun. 2015.
[25] X. Zhang and K. G. Shin, "Delay-optimal broadcast for multihop wireless networks using self-interference cancellation," *IEEE Trans. Mobile Comput.*, vol. 12, no. 1, pp. 7–20, Jan. 2013.

[26] E. Skevakis and I. Lambadaris, "Delay optimal scheduling for network coding broadcast," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.
[27] M. Yu, A. Sprintson, and P. Sadeghi, "On minimizing the average packet decoding delay in wireless network coded broadcast," in *Proc. Int. Symp. Netw. Coding (NetCod)*, Jun. 2015, pp. 1–5.

NAN WANG received the B.S. degree in communication engineering from Liaoning University, in 2002, and the M.S. degree in signal and information processing from Shanxi Normal University, in 2009. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Northeastern University. He has been an Instructor with Shenyang University, since 2010. His research interests include wireless sensor networks, routing algorithm design in ad-hoc networks, and image processing.

HAI ZHAO received the Ph.D. degree in computer application technology from Northeastern University, China, in 1985. He is currently a Professor and a Doctoral Supervisor with the School of Computer Science and Engineering, Northeastern University. He established the International Association for Internet Data Analysis (CAIDA), China's First Node, and maintained long-term research and teaching cooperation with the University of Waterloo, Canada, UTD, USA, and UTA universities. He has applied for ten national patents and software copyrights, and published over 200 articles in related international conferences and journals. His current research interests include embedded systems, sensor networks, body area networks, data and information fusion, computing intelligence, and image processing.



HAIBO JIN received the B.S. and M.S. degrees in computer science and technology from Liaoning Normal University, Dalian, China, in 2006 and 2009, respectively, and the Ph.D. degree in measurement technology and instruments from the Dalian University of Technology, Dalian, China, in 2014. He is currently an Associate Professor with Liaoning Technical University–Huludao. His current research interests are the area of reliability analysis, preventive maintenance, fault prognosis, and network security.

LONG HAI received the B.S. degree in electronic engineering from Shenyang Jianzu University, in 2008, the M.S. degree in electrical and computer engineering from Northeastern University, in 2010, and the Ph.D. degree from the Dalian University of Technology, Dalian, China, in 2016. From September 2012 to October 2013, he was a Visiting Researcher with New York University. From May 2016 to April 2018, he was a Postdoctoral Researcher with Shenzhen University. He is currently an Associate Professor with Shenzhen Polytechnic. His research interests include network coding in lossy networks, routing algorithm design in ad-hoc networks, and secure network coding.