Coded Caching Design for Dynamic Networks

Xianzhang Wu, Minquan Cheng, *Member, IEEE*, Li Chen, *Senior Member, IEEE*, and Congduan Li, *Member, IEEE*

***Abstract*— Coded caching is an effective technique to reduce the data transmission load by exploiting the cache contents across the network. However, most coded caching schemes are designed for static networks that consist of only a placement phase and a delivery phase. In practice, a network maybe dynamic with multiple rounds of placement and delivery phases, and the number of users within the network may vary. In these dynamic networks, a conventional coded caching scheme may lead to the undesired updates at the existing users' cache contents. This paper proposes a centralized coded caching scheme for dynamic networks that can support multiple rounds with newly joining users. It prevents cache contents of the existing users from being updated, extending the service duration of cache devices. Further recognizing the need of information security in coded caching, the considered dynamic networks are featured by two constraints: 1) the library files must be kept secure from a wiretapper who has access to the shared link; 2) any subset of users cannot obtain information from the demands of other users. This consideration leads to another dynamic coded caching scheme that ensures information security. It is shown that the proposed schemes can yield a small subpacketization level and achieve a good rate-memory tradeoff.**

***Index Terms*— Coded caching, content security, dynamic networks, demand privacy, subpacketization level.**

I. Introduction

THE dramatic increase in the use of smart devices leads to an unprecedented growth in internet traffic. This generates a tremendous challenge on smoothing the data transmission over the network, especially during the peak hours. Coded caching has been introduced as an effective technique to alleviate the network pressure by exploiting the cache contents. The original coded caching network [1] consists of a central server which has access to a library of N files of the same size. It provides service to K users over an error free shared link. Each user has a cache memory with a size of M files. A coded caching scheme normally consists of two phases, the placement phase and the delivery phase. In the placement phase, the server sends the properly designed contents to each user's cache without any prior knowledge of the later demands. In the delivery phase, the server is informed with the users' demands. It then broadcasts the coded packets to the users so that each user can reconstruct its desired file with the assistance of their own cache contents. The worst case broadcasting load normalized by the size of file is defined as the transmission rate R, i.e., the minimum number of files that must be communicated so that any possible demand can be satisfied. Under such paradigm, if the packets are cached directly without coding in the placement phase, it is called an uncoded placement; otherwise, it is called a coded placement. We summarize the prior work as follows.

A. Prior Work

Maddah-Ali and Niesen [1] have proposed a coded caching scheme that is realized by the combinatorial uncoded cache placement phase and the network coded delivery phase, which is referred as the MN scheme. It was shown that the transmission rate of the MN scheme is optimal under the constraints of the uncoded placement and $K \leq N$ [2], [3]. For any K and N, a factor of four for the order optimality of the MN scheme was also proved in [4]. Observing that there exist some redundant transmissions in the MN scheme when a file is requested by several users, the authors of [5] designed a scheme that can achieve an optimal transmission rate under the constraint of the uncoded placement. The MN scheme has been extensively studied for several other network scenarios, such as the decentralized caching where each user populates its cache independently of other users [6], [7], the D2D caching networks where users communicate with each other during the delivery phase [8], the combination networks where the server is connected to the users through some intermediate relays [9], the shared cache networks where each user can access one cache and a cache may serve multiple users [10], and the dynamic networks [11], [12], where the uses can freely join or depart the network.

An important parameter for coded caching is subpacketization level, which is defined as the number of packets split in each file. It is known that the implementation complexity of a coded caching scheme increases with the subpacketization

Manuscript received 27 July 2023; revised 11 January 2024 and 28 February 2024; accepted 13 March 2024. Date of publication 19 March 2024; date of current version 16 August 2024. This work is sponsored by the National Natural Science Foundation of China (NSFC) with project IDs 62071498, 62061004, 61901534, 62271514, and U21A20474, the Science, Technology and Innovation Commission of Shenzhen Municipality with project ID JCYJ20210324120002007, and the Guangxi Natural Science Foundation with project ID DA035087. An earlier version of this paper was presented in part at the IEEE International Symposium on Information Theory (ISIT), 2023 [DOI: 10.1109/ISIT54713.2023.10206875]. The associate editor coordinating the review of this article and approving it for publication was A. Cohen. *(Corresponding authors: Li Chen; Congduan Li.)*

Xianzhang Wu is with the College of Computer and Information Science, Fujian Agriculture and Forestry University, Fuzhou 350002, China (e-mail: wuxzh@fafu.edu.cn).

Minquan Cheng is with the Key Laboratory of Education Blockchain and Intelligent Technology, Ministry of Education, and the Guangxi Key Laboratory of Multi-Source Information Mining and Security, Guangxi Normal University, Guilin 541004, China (e-mail: chengqinshi@hotmail.com).

Li Chen is with the School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China (e-mail: chenli55@mail.sysu.edu.cn).

Congduan Li is with the School of Electronics and Communication Engineering, Sun Yat-sen University, Shenzhen 518107, China (e-mail: licongd@mail.sysu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCOMM.2024.3379352.

Digital Object Identifier 10.1109/TCOMM.2024.3379352

level. The MN scheme yields a subpacketization level that grows exponentially with the number of users, which makes them impractical for large networks. This problem has been addressed by several approaches, including placement delivery array (PDA) [13], [14], [15], [16], [17], [18], projective geometry [19] and Ruzsa-Szemerédi graphs [20], and etc. However, they usually trade it with the transmission rate. Recently, it has been realized that applying multi-antenna technique in the coded caching networks can provide a substantial performance improvement in terms of the achievable rate and subpacketization level [12], [21], [22], [23], [24]. In particular, with the spatial multiplexing technique, the scheme of [22] offers a full multiplex gains and coding gains for shared cached networks in which the users are equipped with a large number of transmitting antennas. The scheme of [23] achieves an optimal degrees of freedom (DoF) and a linear subpacketization level based on the zero forcing technique. Later in [24], a scheme that yields a smaller subpacketization level was derived under the fashion of cyclic placement.

In recent years, information security of the coded caching have been investigated in [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], and [35]. In particular, the work of [25] considered the security of library files against a wiretapper who has access to the shared link. The proposed scheme utilized security keys that are shared among the users to secure the transmissions. Its transmission rate was proved to be optimal under the constraint of uncoded placement [26]. Another security coded caching network was studied in [27], where each user cannot learn any information about the files other than its requested one. This scheme was designed based on the MN scheme with an additional secret sharing technique of [28]. Demand privacy under broadcast transmissions was examined in [29], where a private scheme with K users and N files can be realized by a non-private scheme with NK users and N files. Moreover, a private scheme using the maximum distance separable (MDS) code was proposed with order optimal transmission rate under the constraint of private multicast [30]. In the recent work of [32], an information security scheme was proposed through the framework of PDA, which accommodates secure delivery and demand privacy against the colluding users. The above information security coded caching problem has also been extended to other network scenarios that include the shared cache networks [36], the D2D networks [34], [37] and the combination networks [38].

B. Paper Contributions

Note that most of the existing work only considers the coded caching design for static networks that consist of only a placement phase and a delivery phase among a constant number of users. In practice, the coded caching network may need to support multiple rounds of placement and delivery phases, during which the new users may join the network and the server does not have any prior knowledge of the new users in the forthcoming rounds. For example, in a video-on-demand system, new users may join and stay for a while at different rounds. In such dynamic networks, a conventional coded caching scheme may lead to the network with frequent updates at the existing users' cache contents. This is caused by the newly joining users. The frequent updates will shorten the service duration of cache devices, resulting in a waste of network resource, especially in the case of few newly joining users. Therefore, it is important to design a coded caching scheme for dynamic networks that can not only yield a large coding gain but also require the minimal cache content updates. This paper considers the design of coded caching schemes for the dynamic networks, aiming to achieve a low transmission rate with a small subpacketization level. The design for dynamic networks with information security constraint is further considered. Our key technical contributions include:

- We formulate a dynamic coded caching network that can support multiple rounds of placement and delivery phases, where new users are joining in each round. In order to minimize the cache content updates in the placement phase and the amount of transmissions in the delivery phase, a new dynamic coded caching scheme (as stated in *Theorem 1*) is proposed through the combinatorial design. Based on the concatenating placement strategy, the coded messages are designed to generate more multicast opportunities between the existing users and the new users. It is shown that the proposed scheme has advantage either in the subpacketization level or transmission rate over the existing schemes. It also generalizes the network model of [11], making it applicable to the scenarios that can support the new users to have different cache sizes.
- Based on the proposed dynamic coded caching scheme, a new dynamic coded caching scheme that can ensure information security (as stated in *Theorem 2*) is further derived. This design incorporates the secret sharing with the proposed dynamic coded caching scheme, utilizing the key superpositions in a similar fashion as in [32]. It is shown that with a negligible cost of cache memory, the proposed information security scheme can yield a similar transmission rate and subpacketization level as the scheme of *Theorem 1*. This indicates that the information security can be realized without increasing any communication cost for the dynamic coded caching networks. This is similar to the cases of [25] and [38] that also ensure information security for other network scenarios.

C. Paper Organization and Notations

The rest of this paper is organized as follows. In Section II, we present the dynamic coded caching network model and its problem formulation. The new dynamic coded caching scheme is proposed in Section III. Section IV further proposes a dynamic coded caching scheme with information security constraint. Performance analysis of the proposed schemes are given in Section V. Finally, Section VI concludes the paper.

Notations: Let calligraphic symbols and bolded lower-case letters denote sets and vectors, respectively. Symbol $\oplus$ denotes the exclusive-or (XOR) operation. Let $\mathbb{Z}_q$ denote the ring of integers modulo q, and $\mathbb{Z}_q^n$ further denote a set of vectors with elements obtained by the n-fold Cartesian product of $\mathbb{Z}_q$, i.e., $\mathbb{Z}_q^n = \{\mathbf{x} = (x_0, x_1, \ldots, x_{n-1}) \mid (x_0, x_1, \ldots, x_{n-1}) \in \mathbb{Z}_q \times \mathbb{Z}_q \times \cdots \times \mathbb{Z}_q\}$. We use $|\cdot|$ to denote the cardinality of

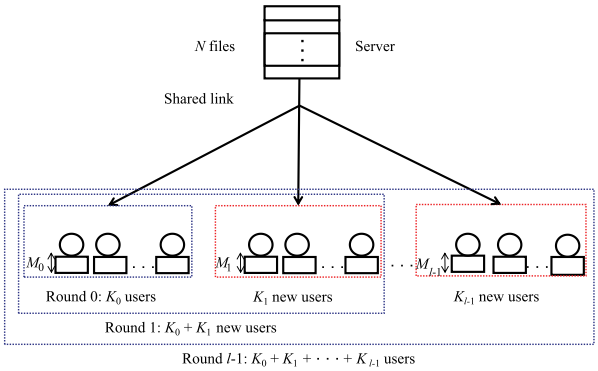

Fig. 1. A $(K_0, K_1, \ldots, K_{l-1}; M_0, M_1, \ldots, M_{l-1}; N)$ dynamic coded caching network.

a set. Let $\mathbb{N}^+$ denote the set of positive integers. The set of consecutive integers is denoted as $[x:y] = \{x, x+1, \ldots, y\}$. For a length-$m$ vector $\mathbf{a}$, let $\mathbf{a}|_i$ denote the ith element of $\mathbf{a}$, where $i \in [0:m-1]$. Finally, the vectors in examples are written as strings, e.g., (1, 1, 1, 1) is written as 1111.

II. Network Model and Problem Formulation

We consider a network model that contains l rounds, where each round consists of a placement phase and a delivery phase. It is illustrated as in Fig. 1. At round j, a server that contains N files of the same size is connected to $\sum_{i=0}^{j} K_i$ users through an error free shared link, where $\sum_{i=0}^{j} K_i \leq N$ and $j \in [0:l-1]$. The N files are denoted as $\mathcal{W} = \{W_0, W_1, \ldots, W_{N-1}\}$, each of size B bits. Let $\mathcal{K}_0$ denote the set of initial users in the network, and $\mathcal{K}_j$ further denote the set of new users at round j. Each user in $\mathcal{K}_j$ is equipped with a dedicated cache with a size of M_j files, where $|\mathcal{K}_j| = K_j$ and $M_j < N$. Note that the cache sizes M_0, M_1, …, M_{l-1} can be different. For clarity, this model is referred as a $(K_0, K_1, \ldots, K_{l-1}; M_0, M_1, \ldots, M_{l-1}; N)$ dynamic coded caching network. It characterizes the dynamic networks at any round j, during which K_j new users join the network and a new coded caching of the existing $\sum_{i=0}^{j-1} K_i$ users is performed. More precisely, let us assume that the network has $\sum_{i=0}^{j-1} K_i$ users at round $j-1$. They are realized by a $(K_0, K_1, \cdots, K_{j-1}; M_0, M_1, \ldots, M_{j-1}; N)$ coded caching scheme. In the forthcoming round j, there are K_j new users joining the network. The server first places the appropriate contents for the $\sum_{i=0}^{j} K_i$ users. Based on their cache contents, another delivery strategy is then conducted so that the coded messages can generate more multicast opportunities between the existing users and the new users. This is different with the one of round $j-1$. That says a new $(K_0, K_1, \cdots, K_j; M_0, M_1, \ldots, M_j; N)$ coded caching scheme is performed. In particular, if $\mathcal{K}_1 = \mathcal{K}_2 = \cdots = \mathcal{K}_{l-1} = \emptyset$, the dynamic coded caching network dissolves into a static network of K_0 users, i.e, a (K_0, M_0, N) coded caching network. If all the users of $\bigcup_{i=1}^{l-1} \mathcal{K}_i$ are viewed as the mobile users, the dynamic coded caching network generalizes the network model of [11] so that the mobile users have different cache sizes.

It is assumed that at the current round, there is no prior knowledge of the forthcoming users. That says information of the number of new users and their cache sizes are not available. At round j, the network operates in two phases as follows.

• **Placement Phase**: Based on a specific cache placement strategy, the server populates the cache of each user k, where $k \in \mathcal{K}_i$, by a cache function $f_k : \mathbb{F}_2^{NB} \mapsto \mathbb{F}_2^{\lfloor M_i B \rfloor}$, where $i \in [0:j]$. This allocation is performed without knowledge of the later demands, and the cache contents of user k is denoted as $\mathcal{Z}_k = f_k(W_0, W_1, \ldots, W_{N-1})$. Note that size of $\mathcal{Z}_k$ cannot be greater than the capacity of the cache memory size.

• **Delivery Phase**: Each user requests an arbitrary file from $\mathcal{W}$. The request vector is denoted by $\mathbf{d} = (d_0, d_1, \ldots, d_{K_0-1}, \ldots, d_{K_0+K_1+\cdots+K_j-1})$, i.e., user k requests file W_{d_k}, where $d_k \in [0:N-1]$. Once the server receives the users' requests, it generates a signal $\mathcal{X}_{\mathbf{d}}$ by using an encoding function $\varphi_{\mathbf{d}} : \mathbb{F}_2^{NB} \mapsto \mathbb{F}_2^{\lfloor RB \rfloor}$, where R is called the transmission rate. The transmitted signal over the shared link is denoted as $\mathcal{X}_{\mathbf{d}} = \varphi_{\mathbf{d}}(W_0, W_1, \ldots, W_{N-1})$. Finally, each user $k \in \mathcal{K}_i$ utilizes a decoding function $\varphi_{\mathbf{d},k}^{-1} : \mathbb{F}_2^{\lfloor RB \rfloor} \times \mathbb{F}_2^{\lfloor M_i B \rfloor} \mapsto \mathbb{F}_2^{B}$, which maps the received signal $\mathcal{X}_{\mathbf{d}}$ and its cache contents $\mathcal{Z}_k$ to compute $\hat{W}_{d_k} \triangleq \varphi_{\mathbf{d},k}^{-1}(X_{\mathbf{d}}, \mathcal{Z}_k)$, which is an estimation of the desired file W_{d_k}. It is required that each user k can reliably recover its desired file, i.e., given an arbitrary small $\varepsilon > 0$, $\max_{\mathbf{d} \in [0:N-1]^\theta} \max_{k \in \bigcup_{i=0}^{j} \mathcal{K}_i} P(\hat{W}_{d_k} \neq W_{d_k}) < \varepsilon$, where $\theta = \sum_{i=0}^{j} K_i$.

In order to minimize the cache content updates, the existing users in the network are assumed to be stationary. They will be involved at the next round with the same placement strategy. This enables their cache contents remain unchanged at the forthcoming rounds. Hence, at the current round, the placement phase should be further partitioned into two subphases, one for the existing users and the other for the new users. Since the cache contents remain unchanged for the existing users, the server would need to further partition the packets that are utilized by the existing users, and meanwhile design the cache placement for the new users. Therefore, the challenge lies in how to design the cache placement for the new users and the multicast messages to ensure a small transmission rate.

III. Dynamic Coded Caching Scheme

This section proposes a dynamic coded caching scheme, which minimizes the cache content updates for the existing users. The design also improves the subpacketization level of the existing scheme. We first introduce our design motivation.

A. Design Motivation

Given a $(K_0, K_1, \ldots, K_{l-1}; M_0, M_1, \ldots, M_{l-1}; N)$ dynamic network, a trivial way of realizing it at round j is to implement a $(K_i; M_i; N)$ coded caching scheme for the users of $\mathcal{K}_i$ separately, where $i \in [0:j]$. However, it may lead to a larger transmission rate since the multicast messages generated for one user group cannot benefit for the other user groups. This is illustrated by the following example, namely as the grouping scheme.

Example 1 (Grouping Scheme): Consider a $(K_0, K_1; M_0, M_1; N)$ dynamic coded caching network with two rounds, where $\mathcal{K}_0 = \{0, 1, 2, 3\}, \mathcal{K}_1 = \{4, 5\}, M_0 = M_1 = 3$ and $N = 6$. At round 1, The users in $\mathcal{K}_0$ and $\mathcal{K}_1$ perform

the $(K_0; M_0; N)$ and $(K_1; M_1; N)$ coded caching schemes, respectively.

• **Placement Phase**: The users in $\mathcal{K}_0$ perform a $(4; 3; 6)$ coded caching scheme. Each file in the server is partitioned into four packets of the same size, i.e., $W_n = \{W_{n,0}, W_{n,1}, W_{n,2}, W_{n,3}\}$, where $n \in [0:5]$. The contents cached by each user in $\mathcal{K}_0$ are $\mathcal{Z}_0 = \{W_{n,0}, W_{n,1}\}$, $\mathcal{Z}_1 = \{W_{n,2}, W_{n,3}\}$, $\mathcal{Z}_2 = \{W_{n,0}, W_{n,2}\}$ and $\mathcal{Z}_3 = \{W_{n,1}, W_{n,3}\}$, where $n \in [0:5]$. Similarly, two new users in $\mathcal{K}_1$ perform another $(2; 3; 6)$ coded caching scheme. Each file in the server is partitioned into two packets of the same size, i.e., $W_n = \{W'_{n,0}, W'_{n,1}\}$, where $n \in [0:5]$. Each user in $\mathcal{K}_1$ caches the following packets, $\mathcal{Z}_4 = \{W'_{n,0}\}$ and $\mathcal{Z}_5 = \{W'_{n,1}\}$, where $n \in [0:5]$.

• **Delivery Phase**: Let us assume that users 0, 1, 2, 3, 4 and 5 request files W_0, W_1, W_2, W_3, W_4 and W_5, respectively. The messages sent by the server consist of two parts. The first part is $W_{0,2} \oplus W_{2,1}$, $W_{1,0} \oplus W_{2,3}$, $W_{0,3} \oplus W_{3,0}$ and $W_{1,1} \oplus W_{3,2}$, which is generated by the $(4; 3; 6)$ coded caching scheme. The second part $W'_{4,1} \oplus W'_{5,0}$, which is generated by the $(2; 3; 6)$ coded caching scheme. Each user can then reconstruct its desired file. The transmission rate is $R(4; 3; 6) + R(2; 3; 6) = 1 + \frac{1}{2} = \frac{3}{2}$.

Note that the MN grouping scheme has a lower transmission rate than that of the grouping scheme in *Example 1*. This is because the transmission rate of the MN scheme is optimal under the constraints of uncoded placement and $K \leq N$. However, its subpacketization level grows exponentially with the number of users. Therefore, as the number of users increases, the subpacketization level of the grouping scheme in *Example 1* will be smaller than that of the MN grouping scheme. Ideally, we would like to design a dynamic coded caching scheme that can yield a small transmission rate, while maintain a low subpacketization level. It can be seen that in *Example 1*, the multicast opportunities between the existing users and new users will be lost, resulting in a larger transmission rate. In order to create more multicast opportunities among all the users at each round, the existing users and the new users should be jointly considered in the design of content delivery, which will be discussed in the next subsection.

B. New Dynamic Coded Caching Scheme

Based on the above observation, this subsection proposes a new dynamic coded caching scheme by concatenating the cache placement of the existing users and the new users so that the coding gains can be enlarged. Different to the construction of [11], in which the multicast messages are generated based on the saturating matching of bipartite graphs, the multicast messages of our proposed scheme are designed through the combinatorial design method. Furthermore, our proposed scheme can support the new users to have different cache sizes and yield a smaller subpacketization level, which is expected to have a wider range of applications. Now we introduce details of the combinatorial cache placement and the content delivery design as follows.

• **Placement Phase**: We focus on the cache placement for the $(K_0, K_1, \ldots, K_{l-1}; M_0, M_1, \ldots, M_{l-1}; N)$ dynamic coded caching network at round j, where $j \in [0 : l-1]$ and $l \in \mathbb{N}^+$. Let $K_i = m_i p_i$ and $M_i = \frac{N z_i}{p_i}$, where m_i, p_i and z_i are positive integers such that $p_i > z_i \geq 1$ and $\lfloor \frac{p_0-1}{p_0-z_0} \rfloor = \lfloor \frac{p_1-1}{p_1-z_1} \rfloor = \cdots = \lfloor \frac{p_j-1}{p_j-z_j} \rfloor$ for $i \in [0 : j]$. For clarity, the users in $\mathcal{K}_i$ are denoted as $\mathcal{K}_i = \{(i, g_i, v_i) \mid (g_i, v_i) \in [0 : m_i - 1] \times [0 : p_i - 1]\}$. Each file in the server is partitioned into $\alpha p_0^{m_0} p_1^{m_1} \cdots p_j^{m_j}$ packets of the same size, i.e.,

$$W_n = \Big\{ W^{(\beta)}_{n,(\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j)} \big| (\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_j) \in \mathbb{Z}_{p_0}^{m_0} \times \mathbb{Z}_{p_1}^{m_1} \times \cdots \times \mathbb{Z}_{p_j}^{m_j}, \beta \in [0 : \alpha - 1] \Big\},$$

where $n \in [0 : N-1]$ and $\alpha = \lfloor \frac{p_0-1}{p_0-z_0} \rfloor = \lfloor \frac{p_1-1}{p_1-z_1} \rfloor = \cdots = \lfloor \frac{p_j-1}{p_j-z_j} \rfloor$. This implies that the subpacketization level of the scheme at round j is $F_j = \alpha p_0^{m_0} p_1^{m_1} \cdots p_j^{m_j}$. The contents cached by user (i, g_i, v_i) are denoted as

$$\mathcal{Z}_{(i,g_i,v_i)} = \Big\{ W^{(\beta)}_{n,(\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j)} \big| (\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_j) \in \mathbb{Z}_{p_0}^{m_0} \times \mathbb{Z}_{p_1}^{m_1} \times \cdots \times \mathbb{Z}_{p_j}^{m_j}, \mathbf{a}_i|_{g_i} \in \{v_i, v_i - 1, \ldots, v_i - (z_i - 1)\}, \beta \in [0 : \alpha - 1], n \in [0 : N - 1] \Big\}, \quad (1)$$

where $(i, g_i, v_i) \in \mathcal{K}_i$. Note that the above computations are performed under modulo p_i. Therefore, each user in $\mathcal{K}_i$ caches a total of $\frac{N F_j z_i}{p_i}$ packets, which requires a cache memory of $\frac{N F_j z_i}{p_i} = M_i F_j$ packets. Moreover, with this concatenating placement strategy, it can be seen that the cache contents of the existing users do not need to be updated at round j. This is because the packets utilized at round j are realized by further partitioning the packets that have been used at round $j-1$.

• **Delivery Phase**: The content delivery at round j can be further described. Suppose that user (i, g_i, v_i) requests file $W_{d_{(i,g_i,v_i)}}$, where $d_{(i,g_i,v_i)} \in [0, N-1]$. Define a multicast function ϕ_i: $\mathbb{Z}_{p_0}^{m_0} \times \mathbb{Z}_{p_1}^{m_1} \times \cdots \times \mathbb{Z}_{p_j}^{m_j} \times [0 : \alpha - 1] \times \mathcal{K}_i \mapsto \mathbb{Z}_{p_0}^{m_0} \times \mathbb{Z}_{p_1}^{m_1} \times \cdots \times \mathbb{Z}_{p_j}^{m_j} \times [0 : p_i - z_i - 1]$, i.e.,

$$\begin{aligned} &\phi_i(\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_j, \beta, i, g_i, v_i) \\ &= (\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{i-1}, a_{i,0}, a_{i,1}, \\ &\quad \ldots, a_{i,g_i-1}, v_i - \beta(p_i - z_i), a_{i,g_i+1} \ldots, a_{i,m_i-1}, \mathbf{a}_{i+1}, \\ &\quad \ldots, \mathbf{a}_j, \mathbf{a}_i|_{g_i} - v_i - 1), \end{aligned} \quad (2)$$

where $\mathbf{a}_i|_{g_i} \notin \{v_i, v_i - 1, \ldots, v_i - (z_i - 1)\}$. Based on (2), for each $\mathbf{b} \in \mathcal{B} = \mathbb{Z}_{p_0}^{m_0} \times \mathbb{Z}_{p_1}^{m_1} \times \cdots \times \mathbb{Z}_{p_j}^{m_j} \times [0 : \max\{p_0 - z_0 - 1, p_1 - z_1 - 1, \ldots, p_j - z_j - 1\}]$, at round j, the server broadcasts the following coded packet to all the users.

$$Y_{\mathbf{b}} = \bigoplus_{\substack{\phi_i(\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j,\beta,i,g_i,v_i)=\mathbf{b}, \beta\in[0:\alpha-1], \\ (\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j)\in\mathbb{Z}_{p_0}^{m_0}\times\mathbb{Z}_{p_1}^{m_1}\times\cdots\times\mathbb{Z}_{p_j}^{m_j}, \\ (g_i,v_i)\in[0:m_i-1]\times[0:p_i-1], i\in[0:j]}} W^{(\beta)}_{d_{(i,g_i,v_i)},(\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j)}. \quad (3)$$

Since $|\mathcal{B}| = p_0^{m_0} p_1^{m_1} \cdots p_j^{m_j} \max\{p_0 - z_0, p_1 - z_1, \ldots, p_j - z_j\}$, the total number of packets sent over the shared link is $R_j F_j = p_0^{m_0} p_1^{m_1} \cdots p_j^{m_j} \max\{p_0 - z_0, p_1 - z_1, \ldots, p_j - z_j\}$.

The above placement and delivery strategy yields the following result that characterizes the property of the proposed dynamic coded caching scheme.

Theorem 1: Given any $p_i, z_i, m_i, l \in \mathbb{N}^+$ with $p_i > z_i \geq 1$ and $\lfloor \frac{p_0-1}{p_0-z_0} \rfloor = \lfloor \frac{p_1-1}{p_1-z_1} \rfloor = \cdots =$

$\lfloor \frac{p_{l-1}-1}{p_{l-1}-z_{l-1}} \rfloor = \alpha$ for $i \in [0 : l-1]$, there exists a $(K_0, K_1, \ldots, K_{l-1}; M_0, M_1, \ldots, M_{l-1}; N)$ dynamic coded caching scheme with $K_i = m_i p_i$ and $M_i = \frac{N z_i}{p_i}$. At round j, the transmission rate of $R_j = \frac{\max\{p_0 - z_0, p_1 - z_1, \ldots, p_j - z_j\}}{\alpha}$ can be achieved with a subpacketization level of $F_j = \alpha p_0^{m_0} p_1^{m_1} \cdots p_j^{m_j}$, where $j \in [0 : l-1]$.

Proof: It is sufficient to prove the decodability at round j, i.e., each user can recover its desired file. In order to show the decodability, the following inequality needs to be proved.

$$\phi_i(\mathbf{a}_0, \ldots, \mathbf{a}_j, \beta, i, g_i, v_i) \neq \phi_i(\mathbf{a}_0, \ldots, \mathbf{a}_j, \beta, i, g_i', v_i') \quad (4)$$

for $(i, g_i, v_i) \neq (i, g_i', v_i')$. Based on (2), if $g_i = g_i'$ and $v_i \neq v_i'$, it can be seen that (4) holds since $v_i - \beta(p_i - z_i) \neq v_i' - \beta(p_i - z_i)$. If $g_i \neq g_i'$, we have $a_{i,g_i} = v_i - \beta(p_i - z_i)$ under the assumption of

$$\phi_i(\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_j, \beta, i, g_i, v_i) = \phi_i(\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_j, \beta, i, g_i', v_i').$$

This implies

$$a_{i,g_i} = v_i - \beta(p_i - z_i) \in \{v_i, v_i - 1, \ldots, v_i - (z_i - 1)\},$$

which contradicts the definition of ϕ_i. Hence, (4) also holds. Similarly, we have

$$\phi_i(\mathbf{a}_0, \ldots, \mathbf{a}_j, \beta, i, g_i, v_i) \neq \phi_{i'}(\mathbf{a}_0, \ldots, \mathbf{a}_j, \beta, i', g_{i'}, v_{i'}) \quad (5)$$

for $i \neq i'$, and

$$\phi_i(\mathbf{a}_0, \ldots, \mathbf{a}_j, \beta, i, g_i, v_i) \neq \phi_i(\mathbf{a}_0', \ldots, \mathbf{a}_j', \beta, i, g_i, v_i') \quad (6)$$

for $v_i \neq v_i'$. It remains to show that

$$\phi_i(\mathbf{a}_0, \ldots, \mathbf{a}_j, \beta, i, g_i, v_i) \neq \phi_i(\mathbf{a}_0', \ldots, \mathbf{a}_j', \beta', i, g_i, v_i) \quad (7)$$

for $(\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_j, \beta) \neq (\mathbf{a}_0', \mathbf{a}_1', \ldots, \mathbf{a}_j', \beta')$. Based on (2), if $(\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_j) = (\mathbf{a}_0', \mathbf{a}_1', \ldots, \mathbf{a}_j')$ and $\beta \neq \beta'$, it can be seen that (7) holds since $v_i - \beta(p_i - z_i) \neq v_i - \beta'(p_i - z_i)$. If $\mathbf{a}_s \neq \mathbf{a}_s'$ for $s \in [0 : j] \backslash \{i\}$, or $a_{i,h} \neq a_{i,h}'$ for $h \in [0, m_i - 1] \backslash \{g_i\}$, based on (2), (7) will hold. If $a_{i,g_i} \neq a_{i,g_i}'$, it can be seen that (7) also holds since $a_{i,g_i} - v_i - 1 \neq a_{i,g_i}' - v_i - 1$.

Without loss of generality, it is assumed that there exist $(\mathbf{a}_0', \mathbf{a}_1', \ldots, \mathbf{a}_j') \in \mathbb{Z}_{p_0}^{m_0} \times \mathbb{Z}_{p_1}^{m_1} \times \cdots \times \mathbb{Z}_{p_j}^{m_j}$ and $\beta' \in [0 : \alpha - 1]$ such that $\phi_i(\mathbf{a}_0', \mathbf{a}_1', \ldots, \mathbf{a}_j', i, g_i, v_i, \beta') = \mathbf{b}$, i.e., the packet $W^{(\beta')}_{d_{(i,g_i,v_i)}, (\mathbf{a}_0', \mathbf{a}_1', \ldots, \mathbf{a}_j')}$ is required by user (i, g_i, v_i). If there exist another $(\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_j) \in \mathbb{Z}_{p_0}^{m_0} \times \mathbb{Z}_{p_1}^{m_1} \times \cdots \times \mathbb{Z}_{p_j}^{m_j}$ and $\beta \in [0 : \alpha - 1]$ satisfying $\phi_i(\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_j, i, g_i', v_i', \beta) = \mathbf{b}$, based on (4), (5) and (7), we have $(i, g_i, v_i) \neq (i, g_i', v_i')$ and $(\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_j, \beta) \neq (\mathbf{a}_0', \mathbf{a}_1', \ldots, \mathbf{a}_j', \beta')$. Therefore, in order to prove the decodability, the following two cases need to be considered.

Case 1: $g_i \neq g_i'$. Based on (2), we have $a_{i,g_i} = v_i - \beta'(p_i - z_i)$. This implies $\mathbf{a}_i|_{g_i} \in \{v_i, v_i - 1, \ldots, v_i - (z_i - 1)\}$, i.e., the packet $W^{(\beta)}_{d_{(i,g_i',v_i')}, \mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_j}$ has been cached by user (i, g_i, v_i).

Case 2: $g_i = g_i'$ and $v_i \neq v_i'$. Based on (6), it can been that $\beta \neq \beta'$. This implies $p_i < 2z_i$. Assume that $\mathbf{a}_i|_{g_i} \notin \{v_i, v_i - 1, \ldots, v_i - (z_i - 1)\}$, i.e., there exists an integer $r \in [1 : p_i - z_i]$ satisfying $\mathbf{a}_i|_{g_i} = v_i + r$. Based on (2), we obtain $v_i = v_i' + (\beta' - \beta)(p_i - z_i)$. As a result, we have

$$\mathbf{a}_i|_{g_i} - v_i' = (\beta' - \beta)(p_i - z_i) + r. \quad (8)$$

This is impossible for $\beta' > \beta$. Note that $p_i - z_i + 1 < (p_i - z_i)(\beta' - \beta) + r < p_i$ and $\mathbf{a}_i|_{g_i} - v_i' < p_i - z_i + 1$ under the fact $\beta', \beta \in [0 : \alpha - 1]$ and $p_i < 2z_i$. It can also be seen that (8) does not hold for $\beta' < \beta$. Therefore, we have $\mathbf{a}_i|_{g_i} \in \{v_i, v_i - 1, \ldots, v_i - (z_i - 1)\}$, i.e., the packet $W^{(\beta)}_{d_{(i,g_i',v_i')}, \mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_j}$ has been cached by user (i, g_i, v_i).

Finally, let us assume there exist $(\mathbf{a}_0'', \mathbf{a}_1'', \ldots, \mathbf{a}_j'') \in \mathbb{Z}_{p_0}^{m_0} \times \mathbb{Z}_{p_1}^{m_1} \times \cdots \times \mathbb{Z}_{p_j}^{m_j}$ and $\beta'' \in [0 : \alpha - 1]$ satisfying $\phi_{i'}(\mathbf{a}_0'', \mathbf{a}_1'', \ldots, \mathbf{a}_j'', i', g_{i'}, v_{i'}, \beta'') = \mathbf{b}$, where $i \neq i'$. With a similar argument of *Case 1*, it can be seen that the packet $W^{(\beta'')}_{d_{(i',g_{i'},v_{i'})}, (\mathbf{a}_0'', \mathbf{a}_1'', \ldots, \mathbf{a}_j'')}$ has been cached by user (i, g_i, v_i). Therefore, its desired packet $W^{(\beta')}_{d_{(i,g_j,v_j)}, (\mathbf{a}_0', \mathbf{a}_1', \ldots, \mathbf{a}_j')}$ can be obtained from the received coded packet $Y_{\mathbf{b}}$,

$$\begin{aligned} Y_{\mathbf{b}} &= W^{(\beta')}_{d_{(i,g_i,v_i)}, (\mathbf{a}_0', \mathbf{a}_1', \ldots, \mathbf{a}_j')} \\ &\oplus \left(\bigoplus_{\substack{\phi_s(\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_j, \beta, s, g_s, v_s) = \mathbf{b}, \\ (\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_j) \in \mathbb{Z}_{p_0}^{m_0} \times \mathbb{Z}_{p_1}^{m_1} \times \cdots \times \mathbb{Z}_{p_j}^{m_j}, \\ (g_s, v_s) \in [0 : m_s - 1] \times [0 : p_s - 1], \\ s \in [0 : j] \backslash \{i\}, \beta \in [0 : \alpha - 1]}} W^{(\beta)}_{d_{(s,g_s,v_s)}, (\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_j)} \right) \\ &\oplus \left(\bigoplus_{\substack{\phi_i(\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_j, \beta, i, g_i', v_i') = \mathbf{b}, \\ (\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_j) \in \mathbb{Z}_{p_0}^{m_0} \times \mathbb{Z}_{p_1}^{m_1} \times \cdots \times \mathbb{Z}_{p_j}^{m_j}, \\ (g_i', v_i') \in [0 : m_i - 1] \times [0 : p_i - 1] \backslash \{(g_i, v_i)\}, \\ \beta \in [0 : \alpha - 1]}} W^{(\beta)}_{d_{(i,g_i',v_i')}, (\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_j)} \right), \end{aligned}$$

since the last two terms can be available in its own cache. Similarly, the other desired packets of user (i, g_i, v_i) can also be recovered. ■

The following example further demonstrates through jointly optimizing the placement and delivery strategy, *Theorem 1* can yield a smaller transmission rate.

Example 2: Consider the same dynamic network as in *Example 1*, i.e., $p_0 = p_1 = 2$, $z_0 = z_1 = 1$, $m_0 = 2$ and $m_1 = 1$. Based on the above placement and delivery design, at round 1, the network performs two phases as follows.

• **Placement Phase**: Each packet $W^{(0)}_{n,(\mathbf{a}_0)}$ used at round 0 is further partitioned into two packets, i.e.,

$$\begin{aligned} W_n &= \left\{ W^{(0)}_{n,(\mathbf{a}_0)} \middle| \mathbf{a}_0 \in \mathbb{Z}_2^2 \right\} = \left\{ W^{(0)}_{n,(\mathbf{a}_0,\mathbf{a}_1)} \middle| (\mathbf{a}_0, \mathbf{a}_1) \in \mathbb{Z}_2^3 \right\} \\ &= \left\{ W^{(0)}_{n,(00,0)}, W^{(0)}_{n,(00,1)}, W^{(0)}_{n,(01,0)}, W^{(0)}_{n,(01,1)}, W^{(0)}_{n,(10,0)}, \right. \\ &\quad \left. W^{(0)}_{n,(10,1)}, W^{(0)}_{n,(11,0)}, W^{(0)}_{n,(11,1)} \right\}, \end{aligned}$$

where $n \in [0 : 5]$. Based on (1), the contents cached by the users of $\mathcal{K}_0 = \{000, 001, 010, 011\}$ remain unchanged as before, i.e.,

$$\begin{aligned} \mathcal{Z}_{000} &= \left\{ W^{(0)}_{n,(00,0)}, W^{(0)}_{n,(00,1)}, W^{(0)}_{n,(01,0)}, W^{(0)}_{n,(01,1)} \right\}, \\ \mathcal{Z}_{001} &= \left\{ W^{(0)}_{n,(10,0)}, W^{(0)}_{n,(10,1)}, W^{(0)}_{n,(11,0)}, W^{(0)}_{n,(11,1)} \right\}, \\ \mathcal{Z}_{010} &= \left\{ W^{(0)}_{n,(00,0)}, W^{(0)}_{n,(00,1)}, W^{(0)}_{n,(10,0)}, W^{(0)}_{n,(10,1)} \right\}, \\ \mathcal{Z}_{011} &= \left\{ W^{(0)}_{n,(01,0)}, W^{(0)}_{n,(01,1)}, W^{(0)}_{n,(11,0)}, W^{(0)}_{n,(11,1)} \right\}, \end{aligned}$$

where $n \in [0:5]$. The contents cached by two new users in $\mathcal{K}_1 = \{100, 101\}$ are

$$\mathcal{Z}_{100} = \left\{W^{(0)}_{n,(00,0)}, W^{(0)}_{n,(01,0)}, W^{(0)}_{n,(10,0)}, W^{(0)}_{n,(11,0)}\right\},$$
$$\mathcal{Z}_{101} = \left\{W^{(0)}_{n,(00,1)}, W^{(0)}_{n,(01,1)}, W^{(0)}_{n,(10,1)}, W^{(0)}_{n,(11,1)}\right\}.$$

• **Delivery Phase**: Let us assume that users $000, 001, 010, 011, 100$ and 101 request files W_0, W_1, W_2, W_3, W_4 and W_5, respectively. Based on (3), the message corresponding to $\mathbf{b} = 0000$ generated by the server will be $Y_{0000} = W^{(0)}_{0,(10,0)} \oplus W^{(0)}_{2,(01,0)} \oplus W^{(0)}_{4,(00,1)}$. This is because $\phi_0(10,1,0,0,0,0) = \phi_0(01,1,0,0,1,0) = \phi_1(00,0,0,1,0,1) = 0000$. Similarly, the other seven messages sent by the server are

$$\begin{aligned}
Y_{0010} &= W^{(0)}_{0,(10,1)} \oplus W^{(0)}_{2,(01,1)} \oplus W^{(0)}_{5,(00,0)},\\
Y_{0100} &= W^{(0)}_{0,(11,0)} \oplus W^{(0)}_{3,(00,0)} \oplus W^{(0)}_{4,(01,1)},\\
Y_{0110} &= W^{(0)}_{0,(11,1)} \oplus W^{(0)}_{3,(00,1)} \oplus W^{(0)}_{5,(01,0)},\\
Y_{1000} &= W^{(0)}_{1,(00,0)} \oplus W^{(0)}_{2,(11,0)} \oplus W^{(0)}_{4,(10,1)},\\
Y_{1010} &= W^{(0)}_{1,(00,1)} \oplus W^{(0)}_{2,(11,1)} \oplus W^{(0)}_{5,(10,0)},\\
Y_{1100} &= W^{(0)}_{1,(01,0)} \oplus W^{(0)}_{3,(10,0)} \oplus W^{(0)}_{4,(11,1)},\\
Y_{1110} &= W^{(0)}_{1,(01,1)} \oplus W^{(0)}_{3,(10,1)} \oplus W^{(0)}_{5,(11,0)}.
\end{aligned}$$

Each user can then reconstruct its desired file. E.g., user 000 requires W_0 and it has cached $W^{(0)}_{0,(00,0)}$,$W^{(0)}_{0,(00,1)}$,$W^{(0)}_{0,(01,0)}$ and $W^{(0)}_{0,(01,1)}$. It can obtain $W^{(0)}_{0,(10,0)}$, $W^{(0)}_{0,(10,1)}$,$W^{(0)}_{0,(11,0)}$ and $W^{(0)}_{0,(11,1)}$ with the following received coded packets Y_{0000}, Y_{0010}, Y_{0100} and Y_{0110}, where $W^{(0)}_{2,(01,0)}$, $W^{(0)}_{4,(00,1)}$, $W^{(0)}_{2,(01,1)}$, $W^{(0)}_{5,(00,0)}$, $W^{(0)}_{3,(00,0)}$, $W^{(0)}_{4,(01,1)}$, $W^{(0)}_{3,(00,1)}$ and $W^{(0)}_{5,(01,0)}$ have been cached. Hence, the transmission rate is $R(4,2;3,3;6) = \frac{8}{8} = 1$, which is smaller than that of $\frac{3}{2}$ in *Example 1*.

Note that the dynamic coded caching scheme characterized in *Theorem 1* can also support departing users. In fact, there are two approaches in realizing this. The first one is to view the departing users as virtual users. E.g., let us assume that the network initially has K_0 users with a cache size of M_0 files. In the first round, K_1 users with a cache size of M_1 files join the network. In the following round 2, K_2 users with a cache size of M_2 files join the network, and K'_1 users of $\mathcal{K}_1$ depart the network. Consequently, the network can be realized by a $(K_0, K_1, K_2; M_0, M_1, M_2; N)$ dynamic coded caching scheme, where the K'_1 departing users of $\mathcal{K}_1$ are considered as virtual users. The second approach is to consider the users of $\mathcal{K}_1 \cup \mathcal{K}_2$ as mobile users, which is similar with the work of [11]. In this case, the above network can be realized by a $(K_0, K_1 - K'_1, K_2; M_0, M'_1, M_2; N)$ dynamic coded caching scheme. It can be seen that the second solution for handling the departing users may perform better. This is because it is realized by a new design of cache placement and content delivery for the remaining users so that the multicasting opportunities can be maximized. However, the second case will inevitably lead to a frequent update at some existing users' cache contents.

IV. New Dynamic Coded Caching Scheme With Information Security

This section further proposes a dynamic coded caching scheme that can ensure information security. It is shown that cache content updates of the existing users can be neglected when the number of library files is large. Let $\tilde{\mathcal{K}}$ denote a subset of users at round j, and $\tilde{\mathcal{K}}^c$ further denote the complementary set of $\tilde{\mathcal{K}}$, i.e., $\tilde{\mathcal{K}}^c = \bigcup_{i=0}^{j} \mathcal{K}_i \backslash \tilde{\mathcal{K}}$. Then, two sufficient conditions for dynamic networks with information security at round j are characterized as follows.

A. Content Security

Any external wiretapper who observes the transmitted signals during the delivery phase cannot obtain any information about the files in the library, i.e.,

$$I(X_{\mathbf{d}}; \mathcal{W}) = 0, \tag{9}$$

where $X_{\mathbf{d}}$ is the transmitted signal over the shared link.

B. Demand Privacy

Any subset of users cannot obtain any information on the demands of other users, i.e.,

$$I\left(\mathbf{d}_{\tilde{\mathcal{K}}^c}; X_{\mathbf{d}}, \mathcal{Z}_{\tilde{\mathcal{K}}}, \mathcal{W} | \mathbf{d}_{\tilde{\mathcal{K}}}\right) = 0, \tag{10}$$

where $\mathbf{d}_{\tilde{\mathcal{K}}} = \{d_k \mid k \in \tilde{\mathcal{K}}\}$, $\mathbf{d}_{\tilde{\mathcal{K}}^c} = \{d_k \mid k \in \tilde{\mathcal{K}}^c\}$ and $\mathcal{Z}_{\tilde{\mathcal{K}}} = \{\mathcal{Z}_k \mid k \in \tilde{\mathcal{K}}\}$.

In order to satisfy the security constraints of (9) and (10), security and privacy keys are placed at the users' caches during the placement phase. These keys are only used to encrypt the transmitted signals once during the delivery phase [39]. This is known as the one-time pad [32]. Based on the placement and delivery strategy introduced in Section III-B, the dynamic networks with requirements (9) and (10) can be realized by the following two phases.

• **Placement Phase**: Each file is partitioned into F_j packets of equal size. The server creates $|\mathcal{B}|$ security keys and $\sum_{i=0}^{j} K_i(1 - \frac{z_i}{p_i})F_j$ privacy keys as follows. The security keys, denoted as $\mathcal{V}^{(j)} = \{V^{(j)}_{\mathbf{b}} \mid \mathbf{b} \in \mathcal{B}\}$, which are randomly generated from $\mathbb{F}_2^{B/F_j}$. In addition, the server generates $\sum_{i=0}^{j} K_i$ random vectors $\mathbf{q}^{(j)}_{(i,g_i,v_i)} = (q^{(j)}_{(i,g_i,v_i),0}, q^{(j)}_{(i,g_i,v_i),1}, \ldots, q^{(j)}_{(i,g_i,v_i),N-1})$ uniformaly over $\mathbb{F}_2^N$. Both the security keys and the random vectors are independent of the library files. Then, the set of privacy keys can be constructed as

$$\begin{aligned}
\mathcal{T}^{(j)} = \Big\{ & T^{(j)}_{(\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j,\beta),(i,g_i,v_i)} \big| \mathbf{a}_i|_{g_i} \notin \{v_i, v_i - 1, \ldots, v_i \\
& - (z_i - 1)\}, \beta \in [0:\alpha - 1], (\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_j) \in \mathbb{Z}_{p_0}^{m_0} \\
& \times \mathbb{Z}_{p_1}^{m_1} \times \cdots \times \mathbb{Z}_{p_j}^{m_j}, (g_i, v_i) \in [0 : m_i - 1] \\
& \times [0 : p_i - 1], i \in [0 : j] \Big\},
\end{aligned} \tag{11}$$

where

$$T^{(j)}_{(\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j,\beta),(i,g_i,v_i)} = \bigoplus_{n \in [0:N-1]} q^{(j)}_{(i,g_i,v_i),n} W^{(\beta)}_{n,(\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j)}.$$

Based on the designed security and privacy keys, each user's cache is populated with two parts of cache contents. The first part is the same as that of the scheme proposed in Section III-B. The second part is some linear combinations of the security and privacy keys, i.e.,

$$\mathcal{Z}''_{(i,g_i,v_i)} = \Big\{ V_{\mathbf{b}}^{(j)} \oplus T^{(j)}_{(\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j,\beta),(i,g_i,v_i)} \big| \phi_i(\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j, \beta,i,g_i,v_i)=\mathbf{b},\mathbf{b}\in\mathcal{B},\beta\in[0:\alpha-1],(\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j)\in\mathbb{Z}_{p_0}^{m_0}\times\mathbb{Z}_{p_1}^{m_1}\times\cdots\times\mathbb{Z}_{p_j}^{m_j}\Big\}, \tag{12}$$

where $(i,g_i,v_i)\in\mathcal{K}_i$. It can be seen that the second part cache contents should be updated at the forthcoming rounds to ensure the content security and demand privacy during the delivery phase. However, the size of $\mathcal{Z}''_{(i,g_i,v_i)}$ is $1-\frac{z_i}{p_i}$ file, which is very small. Its communication cost can be negligible when there are a large number of library files.

• **Delivery Phase**: After receiving the users' demands, the server first generates $\sum_{i=0}^{j} K_i$ vectors as

$$\mathcal{H}_{\mathbf{d}} = \Big\{ \mathbf{h}^{(j)}_{(i,g_i,v_i)} = \Big(h^{(j)}_{(i,g_i,v_i),0}, h^{(j)}_{(i,g_i,v_i),1},\ldots,h^{(j)}_{(i,g_i,v_i),N-1}\Big) \Big| (i,g_i,v_i)\in\mathcal{K}_i, i\in[0:j]\Big\}, \tag{13}$$

where

$$h^{(j)}_{(i,g_i,v_i),y} = \begin{cases} q^{(j)}_{(i,g_i,v_i),y}, \\ \quad \text{if } y\in[0:N-1]\setminus\{d_{(i,g_i,v_i)}\}; \\ q^{(j)}_{(i,g_i,v_i),y}\oplus 1, \quad \text{if } y = d_{(i,g_i,v_i)}. \end{cases} \tag{14}$$

Based on (13), the server creates $|\mathcal{B}|$ coded packets as $\mathcal{S}_{\mathbf{d}} = \{S_{\mathbf{b}} \mid \mathbf{b}\in\mathcal{B}\}$, where

$$S_{\mathbf{b}} = V^{(j)}_{\mathbf{b}} \oplus \bigoplus_{\substack{\phi_i(\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j,\beta,i,g_i,v_i)=\mathbf{b},\beta\in[0:\alpha-1],\\ (\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j)\in\mathbb{Z}_{p_0}^{m_0}\times\mathbb{Z}_{p_1}^{m_1}\times\cdots\times\mathbb{Z}_{p_j}^{m_j},\\ (g_i,v_i)\in[0:m_i-1]\times[0:p_i-1],i\in[0:j]}} \bigoplus_{n\in[0:N-1]} h^{(j)}_{(i,g_i,v_i),n} W^{(\beta)}_{n,(\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j)}. \tag{15}$$

Then, the server broadcasts the signal $X_{\mathbf{d}} = \mathcal{H}_{\mathbf{d}}\cup\mathcal{S}_{\mathbf{d}}$ to all the users at round j. It can be seen that for a large file size B, the additional transmission rate generated by $\mathcal{H}_{\mathbf{d}}$ is marginalized. Note that the size of $\mathcal{H}_{\mathbf{d}}$ is $N\sum_{i=0}^{j}K_i$ bits, and $\frac{N\sum_{i=0}^{j}K_i}{B}\to 0$ as B increases.

The above two phases yield the following dynamic coded caching scheme that can support content security and demand privacy in the delivery phase.

Theorem 2: Given any $p_i, z_i, m_i, l \in \mathbb{N}^+$ with $p_i > z_i \geq 1$ and $\lfloor\frac{p_0-1}{p_0-z_0}\rfloor = \lfloor\frac{p_1-1}{p_1-z_1}\rfloor = \cdots = \lfloor\frac{p_{l-1}-1}{p_{l-1}-z_{l-1}}\rfloor = \alpha$ for $i\in[0:l-1]$, there exists a $(K_0,K_1,\ldots,K_{l-1};M_0,M_1,\ldots,M_{l-1};N)$ dynamic coded caching scheme that can support content security and demand privacy with $K_i = m_i p_i$ and $M_i = 1+\frac{(N-1)z_i}{p_i}$. At round j, the transmission rate of $R_j = \frac{\max\{p_0-z_0,p_1-z_1,\ldots,p_j-z_j\}}{\alpha}$ can be achieved with a subpacketization level of $F_j = \alpha p_0^{m_0}p_1^{m_1}\cdots p_j^{m_j}$, where $j\in[0:l-1]$.

Proof: Content Security: In order to prove (9), it is sufficient to prove $I(\mathcal{W},\mathbf{d};X_{\mathbf{d}})=0$. Based on the property of mutual information, we have

$$\begin{aligned} I(\mathcal{W},\mathbf{d};X_{\mathbf{d}}) &= I(\mathcal{W},\mathbf{d};\mathcal{H}_{\mathbf{d}},\mathcal{S}_{\mathbf{d}}) \\ &= I(\mathcal{H}_{\mathbf{d}};\mathcal{W},\mathbf{d}) + I(\mathcal{S}_{\mathbf{d}};\mathcal{W},\mathbf{d}|\mathcal{H}_{\mathbf{d}}) \\ &\overset{(a)}{=} 0 + I(\mathcal{S}_{\mathbf{d}};\mathcal{W},\mathbf{d}|\mathcal{H}_{\mathbf{d}}) \\ &\overset{(b)}{=} 0, \end{aligned} \tag{16}$$

where (a) holds since $\mathcal{H}_{\mathbf{d}}$ is independent of $\mathcal{W}$ and $\mathbf{d}$ because the vector $\mathbf{h}^{(j)}_{(i,g_i,v_i)}$ of (13) is uniformly distributed over $\mathbb{F}_2^N$, and (b) holds since $\mathcal{S}_{\mathbf{d}}$ is independent of $\mathcal{W},\mathbf{d}$ and $\mathcal{H}_{\mathbf{d}}$ because the term $V_{\mathbf{b}}^{(j)}$ of (15) is randomly chosen from $\mathbb{F}_2^{B/F_j}$.

Demand Privacy: Given any $\tilde{\mathcal{K}}\subseteq\bigcup_{i=0}^{j}\mathcal{K}_i$ and $\tilde{\mathcal{K}}\neq\emptyset$, we have

$$\begin{aligned} I(\mathbf{d}_{\tilde{\mathcal{K}}^c};X_{\mathbf{d}},\mathcal{Z}_{\tilde{\mathcal{K}}},\mathcal{W}|\mathbf{d}_{\tilde{\mathcal{K}}}) &\leq I(\mathbf{d}_{\tilde{\mathcal{K}}^c};X_{\mathbf{d}},\mathcal{Z}_{\tilde{\mathcal{K}}},\mathcal{H}_{\mathbf{d}},\mathcal{W}|\mathbf{d}_{\tilde{\mathcal{K}}}) \\ &\overset{(a)}{=} I(\mathbf{d}_{\tilde{\mathcal{K}}^c};\mathcal{Z}_{\tilde{\mathcal{K}}},\mathcal{H}_{\mathbf{d}},\mathcal{W}|\mathbf{d}_{\tilde{\mathcal{K}}}) \\ &\overset{(b)}{=} I(\mathbf{d}_{\tilde{\mathcal{K}}^c};\mathcal{Z}_{\tilde{\mathcal{K}}}|\mathcal{H}_{\mathbf{d}},\mathcal{W},\mathbf{d}_{\tilde{\mathcal{K}}}) \\ &= H(\mathcal{Z}_{\tilde{\mathcal{K}}}|\mathcal{H}_{\mathbf{d}},\mathcal{W},\mathbf{d}_{\tilde{\mathcal{K}}}) - \\ &\quad H(\mathcal{Z}_{\tilde{\mathcal{K}}}|\mathcal{H}_{\mathbf{d}},\mathcal{W},\mathbf{d}) \\ &\overset{(c)}{=} H(\mathcal{Z}_{\tilde{\mathcal{K}}}|\mathcal{H}_{\mathbf{d}},\mathcal{W}) - H(\mathcal{Z}_{\tilde{\mathcal{K}}}|\mathcal{H}_{\mathbf{d}},\mathcal{W}) \\ &= 0, \end{aligned} \tag{17}$$

where (a) holds since $X_{\mathbf{d}}$ is a function of $\mathcal{H}_{\mathbf{d}}$ and $\mathcal{W}$, (b) holds since $\mathcal{H}_{\mathbf{d}}$ and $\mathcal{W}$ are independent of the users' demands, and (c) holds since the placement is independent of the users' demands.

Decodability: Suppose that user $(i,g_i,v_i)\in\mathcal{K}_i$ requests file $W_{d_{(i,g_i,v_i)}}$, where $i\in[0:j]$. Based on the cache contents of user (i,g_i,v_i), it just needs to show that user $(i,g_i,v_i)\in\mathcal{K}_i$ can obtain all the packets of

$$\Big\{ W^{(\beta)}_{d_{(i,g_i,v_i)},(\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j)} \mid (\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j)\in\mathbb{Z}_{p_0}^{m_0}\times\mathbb{Z}_{p_1}^{m_1}\times\cdots\times\mathbb{Z}_{p_j}^{m_j}, \mathbf{a}_i|_{g_i}\notin\{v_i,v_i-1,\ldots,v_i-(z_i-1)\}, \beta\in[0:\alpha-1], n\in[0:N-1]\Big\}.$$

Without loss of generality, it is assumed that $\phi_i(\mathbf{a}'_0,\mathbf{a}'_1,\ldots,\mathbf{a}'_j,\beta',i,g_i,v_i)=\mathbf{b}$, i.e., the packet $W^{(\beta')}_{d_{(i,g_i,v_i)},(\mathbf{a}'_0,\mathbf{a}'_1,\ldots,\mathbf{a}'_j)}$ is required by user (i,g_i,v_i) but has not been cached in its own cache. Then, (15) can be written as

$$S_{\mathbf{b}} = W^{(\beta')}_{d_{(i,g_i,v_i)},(\mathbf{a}'_0,\mathbf{a}'_1,\ldots,\mathbf{a}'_j)} \oplus \Bigg(V^{(j)}_{\mathbf{b}} \oplus \bigoplus_{n\in[0:N-1]} q^{(j)}_{(i,g_i,v_i),n} W^{(\beta')}_{n,(\mathbf{a}'_0,\mathbf{a}'_1,\ldots,\mathbf{a}'_j)} \Bigg) \oplus \Bigg(\bigoplus_{\substack{\phi_s(\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j,\beta,s,g_s,v_s)=\mathbf{b},\\ (\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j)\in\mathbb{Z}_{p_0}^{m_0}\times\mathbb{Z}_{p_1}^{m_1}\times\cdots\times\mathbb{Z}_{p_j}^{m_j},\\ (g_s,v_s)\in[0:m_s-1]\times[0:p_s-1],\\ s\in[0:j]\setminus\{i\},\beta\in[0:\alpha-1]}} \bigoplus_{n\in[0:N-1]} h^{(j)}_{(s,g_s,v_s),n} W^{(\beta)}_{n,(\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j)} \Bigg)$$

$$\oplus\Bigg(\bigoplus_{\substack{\phi_i(\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j,\beta,i,g_i',v_i')=\mathbf{b},\\(\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j)\in\mathbb{Z}_{p_0}^{m_0}\times\mathbb{Z}_{p_1}^{m_1}\times\cdots\times\mathbb{Z}_{p_j}^{m_j},\\(g_i',v_i')\in[0:m_i-1]\times[0:p_i-1]\backslash(g_i,v_i),\\\beta\in[0:\alpha-1]}}\ \bigoplus_{n\in[0:N-1]} h^{(j)}_{(i,g_i',v_i'),n} W^{(\beta)}_{n,(\mathbf{a}_0,\mathbf{a}_1,\ldots,\mathbf{a}_j)}\Bigg). \tag{18}$$

Based on (12), it can be seen that the second term of (18) has been cached by user (i,g_i,v_i). Further based on the proof of *Theorem 1* and the fact that user (i,g_i,v_i) can get the coefficient vectors from $\mathcal{H}_{\mathbf{d}}$, user (i,g_i,v_i) can compute the last two terms of (18). Therefore, the packet $W^{(\beta')}_{d_{(i,g_i,v_i)},(\mathbf{a}_0',\mathbf{a}_1',\ldots,\mathbf{a}_j')}$ can be decoded from (18). Similarly, the other desired packets of user (i,g_i,v_i) can also be recovered. ■

Continued from *Example 2*, the following example further illustrates the realization of the scheme characterized in *Theorem 2*.

Example 3: Consider a $(4,2;3,3;6)$ dynamic network as in *Example 2* with information security. Based on the above placement and delivery design, at round 1, the network operates the following two phases.

• **Placement Phase**: The first part of the cache contents of the users in $\mathcal{K}_0\cup\mathcal{K}_1=\{000,001,010,011,100,101\}$ is the same as that of the scheme in *Example 2*. For the second part of the cache contents, the server first generates 8 security keys and 24 privacy keys as follows. The 8 security keys are randomly generated from $\mathbb{F}_2^{B/8}$, denoted as

$$\begin{aligned}\mathcal{V}^{(1)}&=\left\{V^{(1)}_{\mathbf{b}}\middle|\mathbf{b}\in\mathbb{Z}_2^3\times\{0\}\right\}\\&=\left\{V^{(1)}_{0000},V^{(1)}_{0010},V^{(1)}_{0100},V^{(1)}_{1000},V^{(1)}_{0110},V^{(1)}_{1010},V^{(1)}_{1100},V^{(1)}_{1110}\right\}.\end{aligned}$$

Each security key is associated with an element of $\mathcal{B}=\mathbb{Z}_2^3\times\{0\}$. The server generates $K_0+K_1=6$ random vectors $\mathbf{q}^{(1)}_{000}$, $\mathbf{q}^{(1)}_{001}$, $\mathbf{q}^{(1)}_{010}$, $\mathbf{q}^{(1)}_{011}$, $\mathbf{q}^{(1)}_{100}$ and $\mathbf{q}^{(1)}_{101}$ over $\mathbb{F}_2^6$, and each one corresponds to a user. Based on (11), the 24 privacy keys can be written as

$$\begin{aligned}\mathcal{T}^{(1)}=\Big\{&T^{(1)}_{(\mathbf{a}_0,\mathbf{a}_1,0),(i,g_i,v_i)}\big|\mathbf{a}_i|_{g_i}\neq v_i,(\mathbf{a}_0,\mathbf{a}_1)\in\mathbb{Z}_2^3,(g_0,v_0)\\&\in[0:1]\times[0:1],(g_1,v_1)\in\{0\}\\&\times[0:1],i\in[0:1]\Big\},\end{aligned}\tag{19}$$

where

$$T^{(1)}_{(\mathbf{a}_0,\mathbf{a}_1,0),(i,g_i,v_i)}=\bigoplus_{n\in[0:N-1]} q^{(1)}_{(i,g_i,v_i),n}W^{(0)}_{n,(\mathbf{a}_0,\mathbf{a}_1)}.$$

Each user (i,g_i,v_i) caches the following superposition keys

$$\begin{aligned}\Big\{&V^{(1)}_{\mathbf{b}}\oplus T^{(1)}_{(\mathbf{a}_0,\mathbf{a}_1,0),(i,g_i,v_i)}\big|\phi_i(\mathbf{a}_0,\mathbf{a}_1,0,i,g_i,v_i)=\mathbf{b},\\&\mathbf{b}\in\mathbb{Z}_2^3\times\{0\},(\mathbf{a}_0,\mathbf{a}_1)\in\mathbb{Z}_2^3\Big\},\end{aligned}$$

i.e.,

$$\begin{aligned}\mathcal{Z}''_{000}=\Big\{&V^{(1)}_{0000}\oplus T^{(1)}_{(10,0,0),000},V^{(1)}_{0010}\oplus T^{(1)}_{(10,1,0),000},\\&V^{(1)}_{0100}\oplus T^{(1)}_{(11,0,0),000},V^{(1)}_{0110}\oplus T^{(1)}_{(11,1,0),000}\Big\},\\\mathcal{Z}''_{001}=\Big\{&V^{(1)}_{1000}\oplus T^{(1)}_{(00,0,0),001},V^{(1)}_{1010}\oplus T^{(1)}_{(00,1,0),001},\\&V^{(1)}_{1100}\oplus T^{(1)}_{(01,0,0),001},V^{(1)}_{1110}\oplus T^{(1)}_{(01,1,0),001}\Big\},\\\mathcal{Z}''_{010}=\Big\{&V^{(1)}_{0000}\oplus T^{(1)}_{(01,0,0),010},V^{(1)}_{0010}\oplus T^{(1)}_{(01,1,0),010},\\&V^{(1)}_{1000}\oplus T^{(1)}_{(11,0,0),010},V^{(1)}_{1010}\oplus T^{(1)}_{(11,1,0),010}\Big\},\\\mathcal{Z}''_{011}=\Big\{&V^{(1)}_{0100}\oplus T^{(1)}_{(00,0,0),011},V^{(1)}_{0110}\oplus T^{(1)}_{(00,1,0),011},\\&V^{(1)}_{1100}\oplus T^{(1)}_{(10,0,0),011},V^{(1)}_{1110}\oplus T^{(1)}_{(10,1,0),011}\Big\},\\\mathcal{Z}''_{100}=\Big\{&V^{(1)}_{0000}\oplus T^{(1)}_{(00,1,0),100},V^{(1)}_{0100}\oplus T^{(1)}_{(01,1,0),100},\\&V^{(1)}_{1000}\oplus T^{(1)}_{(10,1,0),100},V^{(1)}_{1100}\oplus T^{(1)}_{(11,1,0),100}\Big\},\\\mathcal{Z}''_{101}=\Big\{&V^{(1)}_{0010}\oplus T^{(1)}_{(00,0,0),101},V^{(1)}_{0110}\oplus T^{(1)}_{(01,0,0),101},\\&V^{(1)}_{1010}\oplus T^{(1)}_{(10,0,0),101},V^{(1)}_{1110}\oplus T^{(1)}_{(11,0,0),101}\Big\}.\end{aligned}$$

• **Delivery Phase**: Let us assume that users $000,001,010,011,100$ and 101 request files W_0,W_1,W_2,W_3,W_4 and W_5, respectively. Based on (13) and (15), the server broadcasts the following coded packets and $\mathcal{H}_{\mathbf{d}}=\left\{\mathbf{h}^{(j)}_{000},\mathbf{h}^{(j)}_{001},\mathbf{h}^{(j)}_{010},\mathbf{h}^{(j)}_{011},\mathbf{h}^{(j)}_{100},\mathbf{h}^{(j)}_{101}\right\}$ to the users at round j.

$$\begin{aligned}S_{0000}=&\bigoplus_{n\in[0:5]}h_{000,n}W^{(0)}_{n,(10,0)}\oplus\bigoplus_{n\in[0:5]}h_{010,n}W^{(0)}_{n,(01,0)}\\&\oplus\bigoplus_{n\in[0:5]}h_{100,n}W^{(0)}_{n,(00,1)}\oplus V_{0000},\\S_{0010}=&\bigoplus_{n\in[0:5]}h_{000,n}W^{(0)}_{n,(10,1)}\oplus\bigoplus_{n\in[0:5]}h_{010,n}W^{(0)}_{n,(01,1)}\\&\oplus\bigoplus_{n\in[0:5]}h_{101,n}W^{(0)}_{n,(00,0)}\oplus V_{0010},\\S_{0100}=&\bigoplus_{n\in[0:5]}h_{000,n}W^{(0)}_{n,(11,0)}\oplus\bigoplus_{n\in[0:5]}h_{011,n}W^{(0)}_{n,(00,0)}\\&\oplus\bigoplus_{n\in[0:5]}h_{100,n}W^{(0)}_{n,(01,1)}\oplus V_{0100},\\S_{0110}=&\bigoplus_{n\in[0:5]}h_{000,n}W^{(0)}_{n,(11,1)}\oplus\bigoplus_{n\in[0:5]}h_{011,n}W^{(0)}_{n,(00,1)}\\&\oplus\bigoplus_{n\in[0:5]}h_{101,n}W^{(0)}_{n,(01,0)}\oplus V_{0110},\\S_{1000}=&\bigoplus_{n\in[0:5]}h_{001,n}W^{(0)}_{n,(00,0)}\oplus\bigoplus_{n\in[0:5]}h_{010,n}W^{(0)}_{n,(11,0)}\\&\oplus\bigoplus_{n\in[0:5]}h_{100,n}W^{(0)}_{n,(10,1)}\oplus V_{1000},\\S_{0100}=&\bigoplus_{n\in[0:5]}h_{001,n}W^{(0)}_{n,(00,1)}\oplus\bigoplus_{n\in[0:5]}h_{010,n}W^{(0)}_{n,(11,1)}\\&\oplus\bigoplus_{n\in[0:5]}h_{101,n}W^{(0)}_{n,(10,0)}\oplus V_{0100},\\S_{1100}=&\bigoplus_{n\in[0:5]}h_{001,n}W^{(0)}_{n,(01,0)}\oplus\bigoplus_{n\in[0:5]}h_{011,n}W^{(0)}_{n,(10,0)}\\&\oplus\bigoplus_{n\in[0:5]}h_{100,n}W^{(0)}_{n,(11,1)}\oplus V_{1100},\\S_{1110}=&\bigoplus_{n\in[0:5]}h_{001,n}W^{(0)}_{n,(01,1)}\oplus\bigoplus_{n\in[0:5]}h_{011,n}W^{(0)}_{n,(101)}\\&\oplus\bigoplus_{n\in[0:5]}h_{101,n}W^{(0)}_{n,(11,0)}\oplus V_{1110}.\end{aligned}$$

Each user can then reconstruct its desired file. E.g., user 000 requires W_0 and it has cached $W^{(0)}_{0,(00,0)}, W^{(0)}_{0,(00,1)}, W^{(0)}_{0,(01,0)}$ and $W^{(0)}_{0,(01,1)}$. It can obtain $W^{(0)}_{0,(10,0)}$ with the following received coded packet.

$$\begin{aligned} S_{0000} &= \bigoplus_{n\in[0:5]} h_{000,n} W^{(0)}_{n,(10,0)} \oplus \bigoplus_{n\in[0:5]} h_{010,n} W^{(0)}_{n,(01,0)} \\ &\quad \oplus \bigoplus_{n\in[0:5]} h_{100,n} W^{(0)}_{n,(00,1)} \oplus V_{0000} \\ &= W_{0,(10,0)} \oplus \left(V_{0000} \oplus \bigoplus_{n\in[0:5]} q_{000,n} W^{(0)}_{n,(10,0)} \right) \\ &\quad \oplus \left(\bigoplus_{n\in[0:5]} h_{010,n} W^{(0)}_{n,(01,0)} \oplus \bigoplus_{n\in[0:5]} h_{100,n} W^{(0)}_{n,(00,1)} \right). \end{aligned} \tag{20}$$

This is because the second term of (20) has been cached by user 000, while the last term can be computed since its coefficient vectors and the packets $W^{(0)}_{n,(01,0)}$ and $W^{(0)}_{n,(00,1)} (n \in [0:5])$ are known to user 000. Similarly, the other desired packets $W^{(0)}_{0,(10,1)}$,$W^{(0)}_{0,(11,0)}$ and $W^{(0)}_{0,(11,1)}$ of user 000 can also be recovered.

Finally, it should be pointed out that our proposed schemes only support the memory size M_i under the parameters p_i and z_i satisfying $(\frac{p_i-1}{p_i-z_i}-1)/\alpha \le 1 \le (\frac{p_i-1}{p_i-z_i})/\alpha$ for any $i \in [0:j-1]$ and $\alpha \in \mathbb{N}^+$. However, our combinatorial construction approach is also applicable to any positive integers p_i and z_i, where $p_i > z_i \ge 1$. In particular, when parameters p_i and z_i satisfy the constraints of *Theorem 1* and *Theorem 2*, the proposed constructions can achieve a better subpacketization level.

V. Performance Analyses of the New Schemes

This section analyzes the proposed dynamic coded caching schemes in their capability of achieving a smaller subpacketization level and its improved tradeoff with the transmission rate. We start with comparing the most relevant coded caching scheme that also supports dynamic networks.

A. Comparison Between the Schemes of Theorem 1, [11], and MN Grouping Scheme

We first compare our proposed scheme with the scheme of [11] and the MN grouping scheme, i.e., the MN scheme with grouping. Note that the scheme of [11] can support dynamic networks with an order optimal transmission rate. Its features are further characterized in the following lemma.

Lemma 1 ([11]): Given any $K_i, M_i, N \in \mathbb{N}^+$ with $M_i < N$ and $t_i = \frac{K_i M_i}{N} \in [1 : K_i - 1]$, where $i \in [0:1]$, there exists a $(K_0, K_1; M_0, M_1; N)$ dynamic coded caching scheme with a transmission rate of

$$R'_1 \le \begin{cases} \frac{(K_0-t_0)(t_0t_1+t_0+1)}{t_0t_1(t_0+1)} + \frac{K_1-t_1}{t_1(t_0+1)}, \text{if } M_0 \le M_1; \\ \frac{(K_1-t_1)(t_0t_1+t_1+1)}{t_0t_1(t_1+1)} + \frac{K_0-t_0}{t_0(t_1+1)}, \text{if } M_0 > M_1, \end{cases}$$

and a subpacketization level of $F'_1 = t_0 t_1 \binom{K_0}{t_0}\binom{K_1}{t_1}$.

Given any $p_i, z_i, m_i \in \mathbb{N}^+$ and $l = 2$ with $\lfloor \frac{p_0-1}{p_0-z_0} \rfloor = \lfloor \frac{p_1-1}{p_1-z_1} \rfloor$, $\frac{z_0}{p_0} > \frac{z_1}{p_1}$ and $z_i < p_i$ for $i \in [0:1]$, the round 1 subpacketization level and transmission rate of the scheme characterized by *Theorem 1* can be written as $F_1 = \alpha p_0^{m_0} p_1^{m_1}$ and

$$R_1 = \frac{\max\{p_0 - z_0, p_1 - z_1\}}{\alpha},$$

respectively, where $\alpha = \lfloor \frac{p_0-1}{p_0-z_0} \rfloor = \lfloor \frac{p_1-1}{p_1-z_1} \rfloor$. In comparison, by letting $K_i = m_i p_i$ and $\frac{M_i}{N} = \frac{z_i}{p_i}$ in *Lemma 1*, the subpacketization level and transmission rate of the scheme in *Lemma 1* can be written as

$$F'_1 = \prod_{i=0}^{1} m_i z_i \binom{m_i p_i}{m_i z_i}$$

and

$$R'_1 \le \frac{(p_1-z_1)(m_0m_1z_0z_1 + m_1z_1 + 1)}{m_0z_0z_1(m_1z_1+1)} + \frac{p_0-z_0}{z_0(m_1z_1+1)},$$

respectively. Based on Stirling's Formula that given $m \in \mathbb{N}^+$ and $m \to \infty$, $m! = \sqrt{2\pi m}(\frac{m}{e})^m$, when $m_0, m_1 \to \infty$, we have

$$\begin{aligned} F'_1 &= \prod_{i=0}^{1} m_i z_i \binom{m_i p_i}{m_i z_i} \\ &\approx \prod_{i=0}^{1} m_i z_i \sqrt{\frac{p_i}{2\pi z_i m_i (p_i - z_i)}} \left(\frac{p_i^{p_i}}{z_i^{z_i} (p_i - z_i)^{p_i - z_i}} \right)^{m_i}. \end{aligned}$$

Hence, the subpacketization level ratio between the schemes in *Theorem 1* and *Lemma 1* can be written as

$$\begin{aligned} \frac{F_1}{F'_1} = \frac{\alpha}{m_0m_1z_0z_1} \prod_{i=0}^{1} \Bigg(& \sqrt{\frac{2\pi z_i m_i (p_i - z_i)}{p_i}} \\ & \left(\frac{p_i^{p_i}}{p_i z_i^{z_i} (p_i - z_i)^{p_i - z_i}} \right)^{-m_i} \Bigg). \end{aligned} \tag{21}$$

Since $i \in [0:1]$, based on the binomial expansion, we have

$$\begin{aligned} p_i^{p_i} &= (p_i - z_i)^{p_i} + \binom{p_i}{1} (p_i - z_i)^{p_i - 1} z_i + \cdots + z_i^{p_i} \\ &\ge \binom{p_i}{z_i - 1} (p_i - z_i)^{p_i - z_i + 1} z_i^{z_i - 1} + \binom{p_i}{z_i} (p_i - z_i)^{p_i - z_i} \\ &\quad \cdot z_i^{z_i} + \binom{p_i}{z_i + 1} (p_i - z_i)^{p_i - z_i - 1} z_i^{z_i + 1} \\ &= \binom{p_i}{z_i} (p_i - z_i)^{p_i - z_i} z_i^{z_i} + \binom{p_i}{z_i} (p_i - z_i)^{p_i - z_i} z_i^{z_i} \\ &\quad \left(\frac{p_i - z_i}{p_i - z_i + 1} + \frac{z_i}{z_i + 1} \right) \\ &\ge 2 \binom{p_i}{z_i} (p_i - z_i)^{p_i - z_i} z_i^{z_i}. \end{aligned} \tag{22}$$

Substituting (22) into (21) yields

$$\begin{aligned} \frac{F_1}{F'_1} &\le \frac{\alpha}{m_0m_1z_0z_1} \prod_{i=0}^{1} \sqrt{\frac{2\pi z_i m_i (p_i - z_i)}{p_i}} \left(\frac{2\binom{p_i}{z_i}}{p_i} \right)^{-m_i} \\ &= \frac{\alpha}{z_0z_1} \prod_{i=0}^{1} \sqrt{\frac{2\pi z_i (p_i - z_i)}{m_i p_i}} \left(\frac{2\binom{p_i}{z_i}}{p_i} \right)^{-m_i} \end{aligned}$$

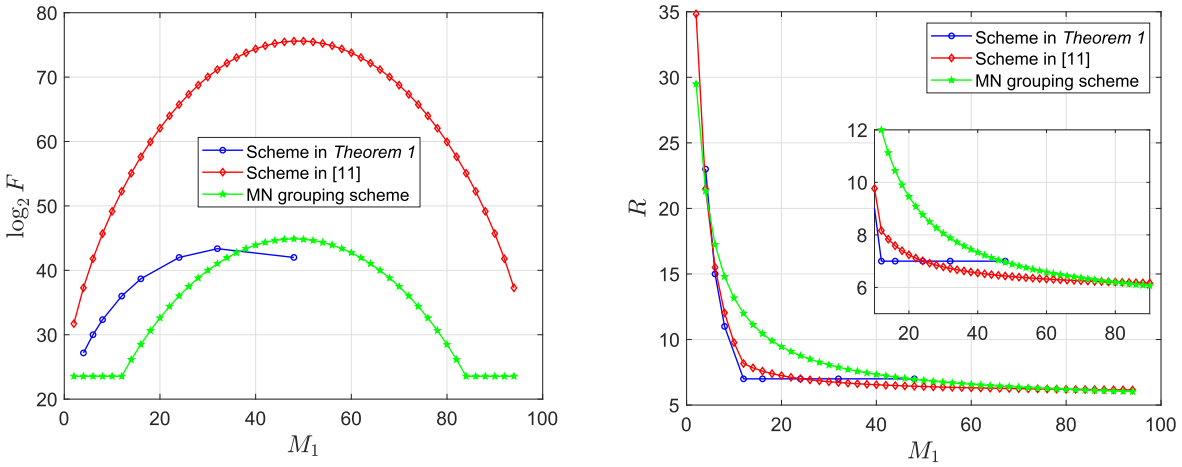

Fig. 2. Comparison between the proposed scheme of *Theorem 1*, the scheme of [11], and MN grouping scheme, where $K_0 = K_1 = 48, M_0 = 12$ and $N = 96$. (a) Subpacketization level; (b) Transmission rate.

$$= \mathcal{O}\left((m_0 m_1)^{-\frac{1}{2}} \eta^{-(m_0+m_1)}\right),$$

where $\eta \geq 2$. Meanwhile, without loss of generality, it is assumed that $p_0 - z_0 < p_1 - z_1$. The transmission rate ratio between the schemes in *Lemma 1* and *Theorem 1* is

$$\frac{R_1'}{R_1} \leq \alpha \left(\frac{m_1 z_1 (m_0 z_0 + 1) + 1}{m_0 z_0 z_1 (m_1 z_1 + 1)} + \frac{p_0 - z_0}{z_0 (m_1 z_1 + 1)(p_1 - z_1)} \right) \approx \frac{\alpha}{z_1},$$

where m_0 and m_1 are two sufficiently large integers. The above analysis shows that for a dynamic network, which is the only situation that the scheme of [11] can support, our proposed scheme is able to substantially reduce the subpacketization level of the scheme of [11]. Meanwhile, the transmission rate of the proposed scheme is at least $\frac{z_1}{\alpha}$ of that of the scheme in [11]. In order to further verify the above comparative characterizations, Fig. 2 compares the subpacketization level and transmission rate between our proposed scheme of *Theorem 1*, the scheme of [11] and the MN grouping scheme. For the proposed scheme, let $p_0 = 8, m_0 = 6, z_0 = z_1 = 1$ and $(m_1, p_1) \in \{(2,24),(3,16),(4,12),(6,8),(8,6),(12,4),(16,3),(24,2)\}$. For the scheme of [11] and the MN grouping scheme, let $K_0 = K_1 = 48, t_0 = 6$ and $t_1 \in [1:47]$. Note that for the subpacketization level of the MN grouping scheme, we choose a larger one between the two groups as a comparison benchmark. It can be seen that with a similar transmission rate, our proposed scheme yields a far smaller subpacketization level than that of [11]. When comparing with the MN grouping scheme, our proposed scheme yields a smaller transmission rate, but at the cost of a slightly increased subpacketization level. It should be noted that when the cache sizes are greater than 37, our proposed scheme has advantages in both the subpacketization level and the transmission rate.

B. Comparison Between the Schemes of Theorem 1 and [7]

We now numerically compare our proposed scheme with the decentralized sequential scheme of [7]. Its features are summarized as follows.

Lemma 2 [7]: Given any $K, K', M, N \in \mathbb{N}^+$ with $M < N$, $K' > 1$ and $t = \frac{K'M}{N} \in [1 : K' - 1]$, where K' is the cardinality of cache content base, there exists a $(K; M; N)$ decentralized sequential coded caching scheme with a transmission rate of

$$R = \begin{cases} \lceil \frac{K}{K'} \rceil \frac{K'-t}{1+t} - \frac{K'-t}{1+t} \prod_{i=0}^{x} \frac{K'-t-1-i}{K'-i}, \\ \quad \text{if } y \leq K'-t; \\ \lceil \frac{K}{K'} \rceil \frac{K'-t}{1+t}, \quad \text{if } y > K'-t, \end{cases}$$

and a subpacketization level of $F = \binom{K'}{t}$, where $x = K' - \lceil \frac{K}{K'} \rceil K' + K - 1$ and $y = K - K'(\lceil \frac{K}{K'} \rceil - 1) + 1$.

Note that the cache size of the scheme in [7] is the same. In order to compare both the subpacketization level and the transmission rate of our proposed scheme with that of [7], we consider the scheme of *Theorem 1* maintaining the same user cache size at round 1. For illustration, let $p_0 = p_1 = 6$, $m_0 = m_1 = 10$ and $z_0 = z_1 \in [1:5]$ for the scheme in *Theorem 1*; Let $K = 120$, $K' = 30$, $t \in [1:29]$ and $K = 120$, $K' = 60$, $t \in [1:59]$ for the scheme in [7]. Fig.3 compares their subpacketization level F and transmission rate R against the cache size M. It can be seen that when $K' = 30$, our proposed scheme in *Theorem 1* has a lower transmission rate than the scheme of [7], but it is realized at the cost of a higher subpacketization level. Meanwhile, when $K' = 60$, our proposed scheme yields a smaller subpacketization level, but with a slightly higher transmission rate when $40 \leq M \leq 80$.

We further demonstrate the advantage of our proposed scheme at round 1 in Table I. For simplicity, the schemes in *Theorem 1* and [7] are parameterized by $(m_0, m_1, p_0, p_1, z_0, z_1, N)$ and (K, K', t, N), respectively. Table I shows that in comparison with the scheme of [7], our proposed scheme has advantage in both the transmission rate and subpacketization level. Furthermore, it should be emphasized that our proposed scheme can support the new users with distinct cache sizes, which is expected to have a wider range of applications.

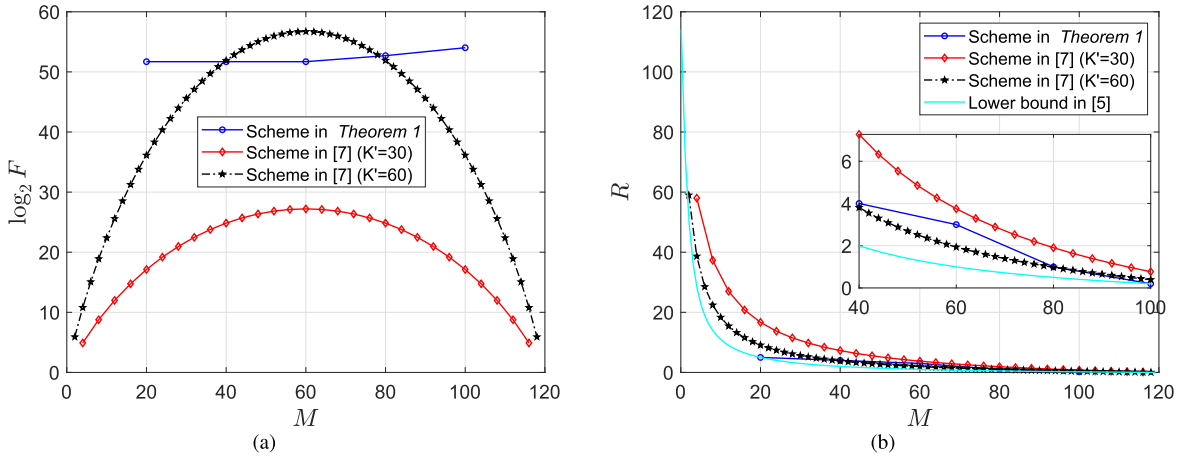

Fig. 3. Comparison between the proposed scheme of *Theorem 1* and the scheme of [7], where $K = K_0 + K_1 = 120$ and $N = 120$. (a) Subpacketization level; (b) Transmission rate.

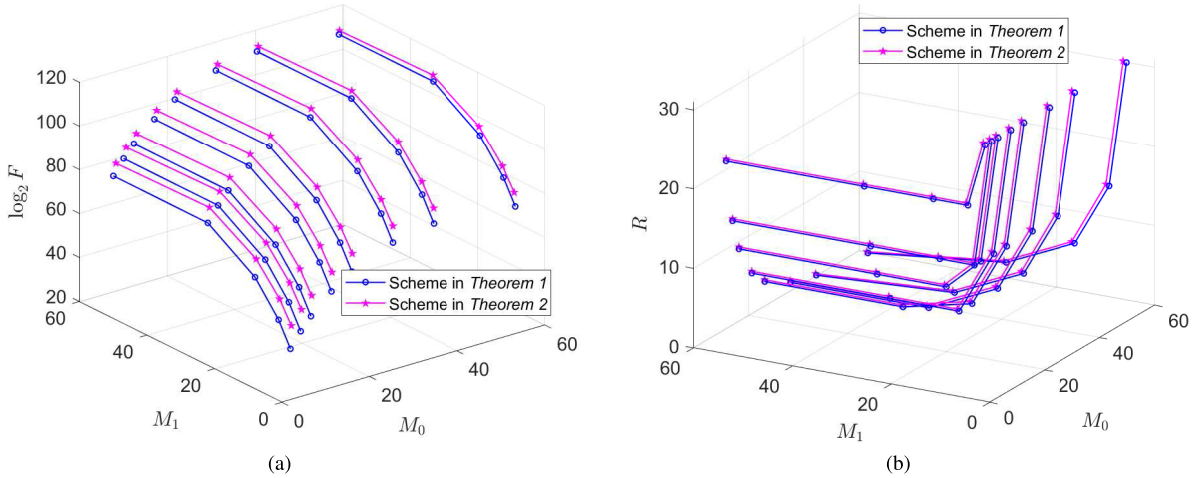

Fig. 4. Comparison between the scheme of *Theorem 1* and the scheme of *Theorem 2*, where $K_0 = 48$ and $K_1 = 64$. (a) Subpacketization level; (b) Transmission rate.

TABLE I

COMPARISON BETWEEN THE SCHEME IN THEOREM 1 AND THE SCHEME IN [7]

Schemes	Parameters	K	F	M	R
$(m_0, m_1, p_0, p_1, z_0, z_1, N)$ in *Theorem 1*	$(10, 5, 2, 2, 1, 30)$	30	32768	15.0	1.00
(K, K', t, N) in [7]	$(30, 20, 10, 30)$	30	184756	15.0	1.82
$(m_0, m_1, p_0, p_1, z_0, z_1, N)$ in *Theorem 1*	$(10, 5, 3, 3, 1, 45)$	45	14348907	15.0	2.00
(K, K', t, N) in [7]	$(45, 30, 10, 45)$	45	30045015	15.0	3.64
$(m_0, m_1, p_0, p_1, z_0, z_1, N)$ in *Theorem 1*	$(10, 2, 3, 3, 2, 36)$	36	1594323	24.0	0.33
(K, K', t, N) in [7]	$(36, 30, 20, 36)$	36	30045015	24.0	0.95
$(m_0, m_1, p_0, p_1, z_0, z_1, N)$ in *Theorem 1*	$(6, 4, 5, 5, 3, 50)$	50	19531250	30.0	1.00
(K, K', t, N) in [7]	$(50, 30, 18, 50)$	50	86493225	30.0	1.26

C. Comparison Between the Schemes of Theorem 1 and Theorem 2

Given any $p_i, z_i, m_i, l \in \mathbb{N}^+$ for $i \in [0 : l-1]$, the cache size of the scheme of *Theorem 2* will be at most one file larger than that of the scheme of *Theorem 1*. In particular, when there are a large number of library files, it can be negligible. In the following we numerically compare the performance of the schemes of *Theorems 1* and *2* with two rounds. For the schemes of *Theorems 1* and *2*, let $(m_0, p_0) \in \{(2, 24), (3, 16), (4, 12), (6, 8), (8, 6), (12, 4), (16, 3), (24, 2)\}$, $(m_1, p_1) \in \{(2, 32), (4, 16), (8, 8), (16, 4), (32, 2)\}$, $z_0 = z_1 = 1$ and $N = 112$,. Fig. 4 shows that with a negligible cost of cache memory, the proposed scheme in *Theorem 2* yields the same transmission rate and subpacketization level as the scheme of *Theorem 1*.

D. Comparison Between the Schemes of Theorem 2 and [29], [31]

Finally, we numerically compare the proposed scheme of *Theorem 2* with the existing information security scheme summarized in Table II.

Note that the security schemes in [29] and [31] only support the networks within one round. In order to compare their performances with our proposed scheme, let $p_0 = 4$, $m_0 = 8$ and $z_0 \in [1 : 3]$ for the scheme in *Theorem 2*; Let $k = N = 32$ and $t \in [1 : 32^2 - 1]$ for the scheme in [29]; Let $k = N = 32$ and $t \in [1 : 31]$ for the scheme in [31]. Fig. 5

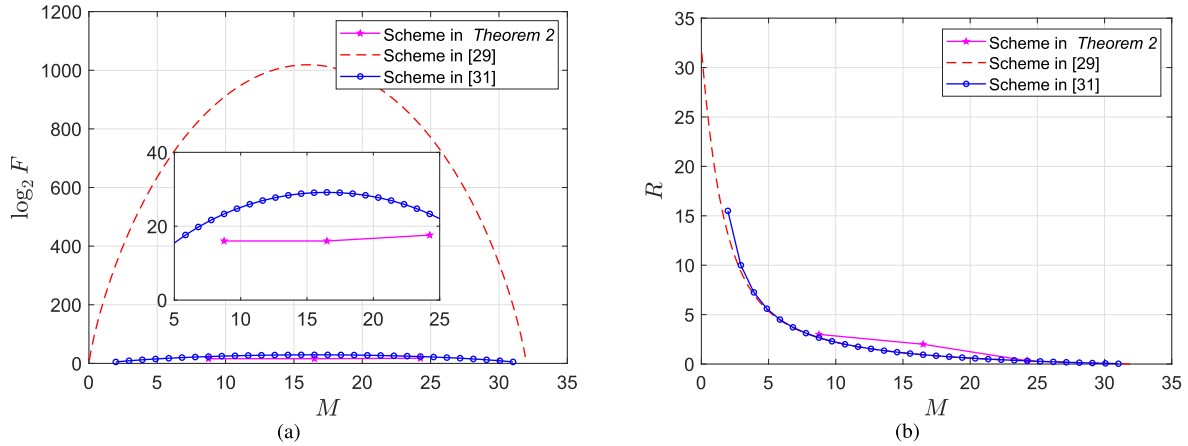

Fig. 5. Comparison between the proposed scheme of *Theorem 2* and the schemes of [29] and [31], where $K = 32$ and $N = 32$. (a) Subpacketization level; (b) Transmission rate.

TABLE II
SUMMARY OF THE EXISTING SECURITY SCHEMES

Schemes and Parameters	K	M	R	F
Scheme in [29], any k, $N \in \mathbb{N}^+$ and $t \in [1 : kN-1]$	k	$\binom{k}{t}$	$\frac{\binom{kN}{t+1}-\binom{(k-1)N}{t+1}}{\binom{kN}{t}}$	$\binom{kN}{t}$
Scheme in [31],any k, $N \in \mathbb{N}^+$ and $t \in [1 : k-1]$	k	$1+\frac{t(N-1)}{k}$	$\frac{\binom{k}{t+1}-\binom{k-\min\{k,N-1\}}{t+1}}{\binom{k}{t}}$	$\binom{k}{t}$

compares their subpacketization level F and transmission rate R against the cache size M. It can be seen that our proposed scheme can yield a smaller subpacketization level, but with a slightly higher transmission rate.

VI. CONCLUSION

This paper has investigated the design of coded caching scheme for dynamic networks that can support multiple rounds of placement and delivery phases, where new users are joining in each round. In order to minimize the cache content updates in the placement phase and the amount of transmissions in the delivery phase, two new dynamic coded caching schemes have been proposed through the combinatorial design. Based on the concatenating based placement contents, the coded messages we designed can generate more multicast opportunities between the existing users and the new users in the delivery phase. Our analytical and numerical results have both shown that the proposed schemes yield a small subpacketization level and achieve a good rate-memory tradeoff.

REFERENCES

[1] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.
[2] K. Wan, D. Tuninetti, and P. Piantanida, "On the optimality of uncoded cache placement," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Sep. 2016, pp. 161–165.
[3] K. Wan, D. Tuninetti, and P. Piantanida, "An index coding approach to caching with uncoded cache placement," *IEEE Trans. Inf. Theory*, vol. 66, no. 3, pp. 1318–1332, Mar. 2020.
[4] H. Ghasemi and A. Ramamoorthy, "Improved lower bounds for coded caching," *IEEE Trans. Inf. Theory*, vol. 63, no. 7, pp. 4388–4413, Jul. 2017, doi: 10.1109/TIT.2017.2705166.
[5] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *IEEE Trans. Inf. Theory*, vol. 64, no. 2, pp. 1281–1296, Feb. 2018.
[6] 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.
[7] S. Jin, Y. Cui, H. Liu, and G. Caire, "Order-optimal decentralized coded caching schemes with good performance in finite file size regime," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1–7.
[8] 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.
[9] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Caching in combination networks," in *Proc. 49th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2015, pp. 1269–1273.
[10] E. Peter, K. K. Krishnan Namboodiri, and B. Sundar Rajan, "Shared cache coded caching schemes with known user-to-cache association profile using placement delivery arrays," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Mumbai, India, Nov. 2022, pp. 678–683.
[11] Q. Zhang, L. Zheng, M. Cheng, and Q. Chen, "On the dynamic centralized coded caching design," *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 2118–2128, Apr. 2020.
[12] M. Abolpour, M. Salehi, and A. Tölli, "Cache-aided communications in MISO networks with dynamic user behavior: A universal solution," 2023, *arXiv:2304.11623*.
[13] M. Cheng, J. Jiang, Q. Yan, and X. Tang, "Constructions of coded caching schemes with flexible memory size," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4166–4176, Jun. 2019.
[14] Q. Yan, M. Cheng, X. Tang, and Q. Chen, "On the placement delivery array design for centralized coded caching scheme," *IEEE Trans. Inf. Theory*, vol. 63, no. 9, pp. 5821–5833, Sep. 2017.
[15] Q. Yan, X. Tang, Q. Chen, and M. Cheng, "Placement delivery array design through strong edge coloring of bipartite graphs," *IEEE Commun. Lett.*, vol. 22, no. 2, pp. 236–239, Feb. 2018.
[16] X. Wu, M. Cheng, C. Li, and L. Chen, "Design of placement delivery arrays for coded caching with small subpacketizations and flexible memory sizes," *IEEE Trans. Commun.*, vol. 70, no. 11, pp. 7089–7104, Nov. 2022.
[17] C. Shangguan, Y. Zhang, and G. Ge, "Centralized coded caching schemes: A hypergraph theoretical approach," *IEEE Trans. Inf. Theory*, vol. 64, no. 8, pp. 5755–5766, Aug. 2018.
[18] X. Wu, M. Cheng, L. Chen, C. Li, and Z. Shi, "Design of coded caching schemes with linear subpacketizations based on injective arc coloring of regular digraphs," *IEEE Trans. Commun.*, vol. 71, no. 5, pp. 2549–2562, May 2023.
[19] H. H. S. Chittoor, P. Krishnan, K. V. S. Sree, and B. Mamillapalli, "Subexponential and linear subpacketization coded caching via projective geometry," *IEEE Trans. Inf. Theory*, vol. 67, no. 9, pp. 6193–6222, Sep. 2021.
[20] K. Shanmugam, A. M. Tulino, and A. G. Dimakis, "Coded caching with linear subpacketization is possible using ruzsa-Szeméredi graphs," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Aachen, Germany, Jun. 2017, pp. 1237–1241.
[21] E. Parrinello, A. Unsal, and P. Elia, "Fundamental limits of coded caching with multiple antennas, shared caches and uncoded prefetching," *IEEE Trans. Inf. Theory*, vol. 66, no. 4, pp. 2252–2268, Apr. 2020.

[22] E. Parrinello, P. Elia, and E. Lampiris, "Extending the optimality range of multi-antenna coded caching with shared caches," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2020, pp. 1675–1680.
[23] M. Salehi, A. Tölli, and S. P. Shariatpanahi, "A multi-antenna coded caching scheme with linear subpacketization," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.
[24] M. Salehi, E. Parrinello, S. P. Shariatpanahi, P. Elia, and A. Tölli, "Low-complexity high-performance cyclic caching for large MISO systems," *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 3263–3278, May 2022.
[25] 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.
[26] M. Bahrami, M. A. Attia, R. Tandon, and B. Vasic, "Towards the exact rate-memory trade-off for uncoded caching with secure delivery," in *Proc. 55th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Oct. 2017, pp. 878–885.
[27] 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.
[28] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.
[29] S. Kamath, "Demand private coded caching," 2019, *arXiv:1909.03324*.
[30] K. Wan and G. Caire, "On coded caching with private demands," *IEEE Trans. Inf. Theory*, vol. 67, no. 1, pp. 358–372, Jan. 2021.
[31] Q. Yan and D. Tuninetti, "Fundamental limits of caching for demand privacy against colluding users," *IEEE J. Sel. Areas Inf. Theory*, vol. 2, no. 1, pp. 192–207, Mar. 2021.
[32] Q. Yan and D. Tuninetti, "Key superposition simultaneously achieves security and privacy in cache-aided linear function retrieval," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 5250–5263, 2021.
[33] C. Gurjarpadhye, J. Ravi, S. Kamath, B. K. Dey, and N. Karamchandani, "Fundamental limits of demand-private coded caching," *IEEE Trans. Inf. Theory*, vol. 68, no. 6, pp. 4106–4134, Jun. 2022, doi: 10.1109/TIT.2022.3150336.
[34] K. Wan, H. Sun, M. Ji, D. Tuninetti, and G. Caire, "On the fundamental limits of device-to-device private caching under uncoded cache placement and user collusion," *IEEE Trans. Inf. Theory*, vol. 68, no. 9, pp. 5701–5729, Sep. 2022.
[35] K. Wan, M. Cheng, D. Liang, and G. Caire, "Multiaccess coded caching with private demands," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2022, pp. 1390–1395.
[36] S. S. Meel and B. S. Rajan, "Secretive coded caching with shared caches," *IEEE Commun. Lett.*, vol. 25, no. 9, pp. 2849–2853, Sep. 2021.
[37] A. A. Zewail and A. Yener, "Device-to-device secure coded caching," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1513–1524, 2020.
[38] A. A. Zewail and A. Yener, "Combination networks with or without secrecy constraints: The impact of caching relays," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1140–1152, Jun. 2018.
[39] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, no. 4, pp. 656–715, Oct. 1949, doi: 10.1002/j.1538-7305.1949.tb00928.x.

Xianzhang Wu received the B.S. degree in mathematics and applied mathematics from Minjiang University, Fuzhou, China, in 2014, the M.S. degree in applied mathematics from Fuzhou University, Fuzhou, in 2018, and the Ph.D. degree in communication and information system from Sun Yat-sen University, Shenzhen, China, in 2023. He is currently a Lecturer with the College of Computer and Information Science, Fujian Agriculture and Forestry University, Fuzhou. His research interests include combinatorics, graph theory, caching networks, and their interactions.

Minquan Cheng (Member, IEEE) received the Ph.D. degree from the Department of Social Systems and Management, Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki, Japan, in 2012. Then, he joined Guangxi Normal University, Guilin, Guangxi, China, where he is currently a Full Professor with the School of Computer Science and Information Technology. His research interests include combinatorics, coding theory, cryptography, and their interactions.

Li Chen (Senior Member, IEEE) received the B.Sc. degree in applied physics from Jinan University, China, in 2003, and the M.Sc. degree in communications and signal processing and the Ph.D. degree in communications engineering from Newcastle University, U.K., in 2004 and 2008, respectively. From 2007 to 2010, he was a Research Associate with Newcastle University. In 2010, he returned China, as a Lecturer with the School of Information Science and Technology, Sun Yat-sen University, Guangzhou. From 2011 to 2012, he was a Visiting Researcher with the Institute of Network Coding, The Chinese University of Hong Kong, where he was an Associate Professor and a Professor from 2011 and 2016. Since 2013, he has been the Associate Head of the Department of Electronic and Communication Engineering (ECE). From July 2015 to October 2015, he was a Visitor with the Institute of Communications Engineering, Ulm University, Germany. From October 2015 to June 2016, he was a Visiting Associate Professor with the Department of Electrical Engineering, University of Notre Dame, USA. From 2017 to 2020, he was the Deputy Dean of the School of Electronics and Communication Engineering. His research interests include information theory, error-correction codes, and data communications. He is a Senior Member of Chinese Institute of Electronics (CIE). He is a member of the IEEE Information Theory Society Board of Governors and its External Nomination Committee and the Chair of its Conference Committee. He is also the Chair of the IEEE Information Theory Society Guangzhou Chapter. He has been organizing several international conferences and workshops, including the 2018 IEEE Information Theory Workshop (ITW) at Guangzhou and the 2022 IEEE East Asian School of Information Theory (EASIT) at Shenzhen, for which he is the General Co-Chair. He is also the TPC Co-Chair of the 2022 IEEE/CIC International Conference on Communications in China (ICCC) at Foshan. He is an Associate Editor of IEEE TRANSACTIONS ON COMMUNICATIONS.

Congduan Li (Member, IEEE) received the B.S. degree from the University of Science and Technology Beijing, China, in 2008, the M.S. degree from Northern Arizona University, AZ, USA, in 2011, and the Ph.D. degree from Drexel University, PA, USA, in 2015, all in electrical engineering. From October 2015 to August 2018, he was a Post-Doctoral Research Fellow with the Institute of Network Coding, The Chinese University of Hong Kong, and the Department of Computer Science, City University of Hong Kong. He is currently an Associate Professor with the School of Electronics and Communication Engineering, Sun Yat-sen University, China. His research interests include networks, such as coding, security, wireless, storage, and caching.