

Robust, Private and Secure Cache-Aided Scalar Linear Function Retrieval From Coded Servers

Qifa Yan[✉], *Member, IEEE*, and Daniela Tuninetti[✉], *Fellow, IEEE*

Abstract—This work investigates a system where each user aims to retrieve a scalar linear function of the files of a library, which are Maximum Distance Separable coded and stored at multiple distributed servers. The system needs to guarantee *robust decoding* in the sense that each user must decode its demanded function with signals received from any subset of servers whose cardinality exceeds a threshold. In addition, (a) the content of the library must be kept secure from a wiretapper who obtains all the signals from the servers; (b) any subset of users together can not obtain any information about the demands of the remaining users; and (c) the users' demands must be kept private against all the servers even if they collude. Achievable schemes are derived by modifying existing Placement Delivery Array (PDA) constructions, originally proposed for single-server single-file retrieval coded caching systems without any privacy or security or robustness constraints. It is shown that the PDAs describing the original Maddah-Ali and Niesen's coded caching scheme result in a load-memory tradeoff that is optimal to within a constant multiplicative gap, except for the small memory regime when the number of file is smaller than the number of users. As by-products, improved order optimality results are derived for three less restrictive systems in all parameter regimes.

Index Terms—Coded caching, distributed storage, maximum distance separable code, placement delivery array, privacy, robust decoding, scalar linear function retrieval, security.

I. INTRODUCTION

CODED caching, introduced by Maddah-Ali and Niesen (MAN) [2], is a technique to reduce the peak-time communication load across a bottleneck shared link by leveraging the multicast opportunities created by content pre-stored at users' local caches. The model consists of a single server, multiple users, and two phases. In the *placement phase*, the users' caches are populated without the knowledge of their future demands. In the *delivery phase*, when users' demands are revealed, the server satisfies them by transmitting coded

packets over the shared link. For a system with N files and K users, the MAN scheme achieves the optimal load-memory tradeoff among all uncoded placement schemes when $N \geq K$ [3], and for $N < K$ after removing some redundant transmissions [4]. Recently, it was shown that allowing the users to demand arbitrary linear combinations of the files does not increase the load compared to the case single file retrieval, at least under uncoded placement [5].

Content security, demand (both user- and server-side) privacy, and robustness are critical aspects of practical systems.

A. Content Security

In [6], the content of the library must be protected against an external wiretapper who obtains the signals transmitted during the delivery phase. The key idea in [6] is that users cache the same content as in the MAN scheme [2], and in addition also share some *security keys* for the part of the files that were not cached in the MAN scheme. The latter is done in a structured way so that each user can retrieve all the multicast signals it needs to decode.

B. User-Side Demand Privacy

Schemes that guarantee user privacy, that is, no user can infer the demand of another user after the delivery phase, were proposed in [7]. In particular, user privacy can be guaranteed by adding virtual users [7], [8]. We investigated user privacy against colluding users in [9], for both single file retrieval and scalar linear function retrieval, where we imposed that any subset of users must not obtain any information about the demands of other users even if they exchange the content in their caches. The key idea in [9] is that, in addition to the cached contents as in the MAN scheme [2], each user also privately caches some *privacy keys*, which are composed as random linear combinations of the parts of the files that were not cached in the MAN scheme. The demands are added by the same coefficients used to generate the privacy keys, so that each user can decode its demanded files with the privacy keys.

C. Content Security & User-Side Demand Privacy

We investigated simultaneous content Security and user demand Privacy for scalar Linear Function Retrieval (SP-LFR) in [10], where we designed a *key superposition* scheme to guarantee both conditions at once by superposing (i.e., sum together) the security keys and privacy keys. We showed that the load-memory tradeoff in this case is the same as in the setup with only content security guarantees. The idea of key superposition was incorporated into the framework

Manuscript received June 1, 2021; revised November 1, 2021; accepted December 22, 2021. Date of publication January 14, 2022; date of current version February 17, 2022. This work was supported in part by NSF Award under Grant 1910309. An earlier version of this paper was presented at the 2021 IEEE Symposium on Information Theory (ISIT) [1] [DOI: 10.1109/ISIT45174.2021.9518207]. (Corresponding author: Qifa Yan.)

Qifa Yan was with the Electrical and Computer Engineering Department, University of Illinois Chicago, Chicago, IL 60607 USA. He is now with the Information Coding and Transmission Key Laboratory of Sichuan Province, CSNMT International Cooperation Research Centre (MoST), Southwest Jiaotong University, Chengdu 611756, China (e-mail: qifayan@swjtu.edu.cn).

Daniela Tuninetti is with the Electrical and Computer Engineering Department, University of Illinois Chicago, Chicago, IL 60607 USA (e-mail: danielat@uic.edu).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/JSAC.2022.3142360>.

Digital Object Identifier 10.1109/JSAC.2022.3142360

of Placement Delivery Array (PDA), which was known to depict both placement and delivery phases in a single array for coded caching systems with neither security or privacy constraint [11]. The advantage of the PDA framework is that low subpacketization schemes can be obtained directly from existing PDA constructions, such as the ones in [11]–[15].

D. Server-Side Demand Privacy

Server-side demand privacy has been thoroughly investigated for the case of multiple servers and a single user, which is known as the Private Information Retrieval (PIR) problem [16]. The capacity of PIR has been characterized in [17] for single file retrieval, in [18] for scalar linear function retrieval, and in [19] or single file retrieval and coluding servers. PIR with a cache-aided user was investigated in [20]–[23].

E. MDS Coded Servers and Decoding Robustness

Since node failures and erasures commonly arise in storage systems, redundancy is desirable [24]. Maximum Distance Separable (MDS) codes are often used to code the data stored across servers. The advantage of MDS coded servers is that it saves storage while allowing unresponsive servers. PIR from MDS-coded servers has been investigated in [25]–[27], and the capacity was characterized in [25]. The schemes in [26], [27] have almost optimal sub-packetization among all schemes achieving the smallest download rate. The PIR schemes in [28] have asymptotically optimal download rate when any number of unresponsive servers not exceeding some threshold show up.

Notice that, the content security and user-side privacy are mainly investigated in coded caching literature, which typically involves *a single server and multiple users*, while the MDS-coded servers and server-side privacy are mainly investigated in the PIR literature, which typically involves *multiple servers and a single user*. For *multiple-server-multiple-user* systems, the techniques from coded caching and PIR were combined in [29], [30] to guarantee server-side privacy. However, how to satisfy all the above requirement is not clear as far as we know.

1) *Contributions and Paper Organization:* In this paper, we combine all the above mentioned requirements in a system whose model is depicted in Fig. 1. The model consists of H servers, N files, and K users. Each of the N files is stored, as an (H, L) MDS coded version,¹ at all servers. Each server is connected to all the users via a dedicated shared link, but may not be able to reach all the users. The novel aspect of this work is to *design coded caching schemes that are robust to some servers' unavailability*, that is, each user must be able to retrieve an arbitrary scalar linear function of the files from the signals obtained from an arbitrary subset of L servers (out of H servers). The security [6], user-side privacy [9] and server-side privacy [19] conditions are also imposed. We refer to this

¹An (H, L) MDS code encodes L information packets into H coded packets, with the property that upon obtaining any L (out of H) coded packets one can recover the L information packets.

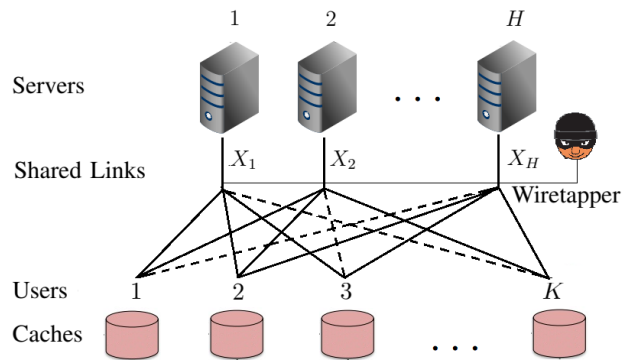


Fig. 1. System model.

model as a Robust Secure and (server- and user-side) Private Linear Function Retrieval (RSP-LFR) problem.

Our key idea on how to guarantee all those conditions simultaneously is to extend the key superposition scheme in [10]. In particular, the technique of superposing user-side privacy and security keys is used in the placement phase, while in the delivery phase, the multicast signals are created in the MDS code domain, where the MDS coded version of the keys are added to the MDS coded multicast signals. Robustness is guaranteed by the linearity property of the MDS code. Security and (server- and user-side) privacy are guaranteed since each transmitted signal is accompanied by an appropriate MDS coded key.

Our main contributions for the proposed RSP-LFR model are as follows.

- 1) We propose a procedure to obtain a RSP-LFR scheme from a given PDA, so that low-subpacketization RSP-LFR schemes can be easily obtained from various existing PDA constructions [11]–[15]. Interestingly, with the same PDA, compared to the single server SP-LFR system in [10], the achieved memory size is the same, but the load is scaled by a factor H/L , i.e., the inverse of the rate of the MDS code used to encode the library files.
- 2) Following the proposed procedure, RSP-LFR schemes based on the PDAs that describe the original MAN scheme in [2] (MAN-PDAs) are proved to achieve the best load-memory tradeoffs among all PDA-based RSP-LFR schemes. Moreover, we show that they have the smallest subpacketization among all PDA based schemes achieving the same load-memory pairs.
- 3) The load-memory tradeoff achieved by MAN-PDAs is proved to be to within a constant multiplicative gap from the optimal load-memory tradeoff, except for the regime of small memory and less files than users.
- 4) For three less restrictive models, where some conditions are dropped, we propose schemes for the corresponding setups that improve the load-memory tradeoffs of the novel MAN-PDA-based RSP-LFR scheme. The idea for improving the tradeoff in less restrictive models is as follows. In the case where security is not imposed, security keys can be removed, and hence, some signals in the delivery phase became redundant and can be

removed akin to [4], [5], [10]. Moreover, those improved schemes are shown to be optimal to within a constant multiplicative gap in their respective setups in all parameter regimes, and the gap is lower than previously known schemes.

The rest of this paper is organized as follows. Section II gives the formal problem definition. Section III reviews the PDA framework and gives an illustrative example. Section IV summarizes our main results, where the proof details are deferred to Sections V–VII. Section VIII presents some numerical results. Section IX concludes the paper.

2) *Notation Convention*: In this paper, \mathbb{N}^+ denotes the set of positive integers; \mathbb{F}_q and \mathbb{F}_q^n denote the finite field of cardinality q , for some prime power q , and the n -dimensional vector space over \mathbb{F}_q , respectively. For two integers m, n such that $m \leq n$, we use $[m : n]$ to denote the set of the first positive integers $\{m, \dots, n\}$; $[1 : n]$ is also denoted by $[n]$ for short. We use $X_{\mathcal{A}}$ to denote the tuple composed of $\{X_i : i \in \mathcal{A}\}$ for some integer set \mathcal{A} , where the elements are ordered increasingly, e.g., $X_{[3]} = (X_1, X_2, X_3)$. For variables with two or more indices, e.g., $X_{i,j}$, we use $X_{\mathcal{A},\mathcal{B}}$ to denote the tuple $\{X_{i,j} : i \in \mathcal{A}, j \in \mathcal{B}\}$, where the elements are listed in lexicographical order, e.g. $X_{[3],[2]} = (X_{1,1}, X_{1,2}, X_{2,1}, X_{2,2}, X_{3,1}, X_{3,2})$.

II. SYSTEM MODEL

Let N, K, L, H be positive integers satisfying $L \leq H$. The (N, K, L, H) RSP-LFR system, illustrated in Fig. 1, consists of H servers (denoted by $1, \dots, H$), where each server is connected to K users (denoted by $1, \dots, K$) via a dedicated shared-link. A file library of N files (denoted by $W_1, \dots, W_N \in \mathbb{F}_q^B$) are stored at the H servers in the form of an (H, L) MDS code as follows, where B denotes the file length. Each file $W_n, n \in [N]$, is composed of L equal-size subfiles $W_{n,1}, \dots, W_{n,L} \in \mathbb{F}_q^{B/L}$ and is encoded into H coded subfiles $\bar{W}_{n,1}, \dots, \bar{W}_{n,H} \in \mathbb{F}_q^{B/L}$ with a given (H, L) MDS code with generator matrix

$$G = \begin{bmatrix} g_{1,1} & \dots & g_{1,H} \\ \vdots & \ddots & \vdots \\ g_{L,1} & \dots & g_{L,H} \end{bmatrix},$$

that is, the coded subfiles are given by

$$\begin{aligned} &(\bar{W}_{n,1}, \dots, \bar{W}_{n,H}) \\ &= \left(\sum_{l \in [L]} g_{l,1} W_{n,l}, \dots, \sum_{l \in [L]} g_{l,H} W_{n,l} \right), \quad \forall n \in [N]. \end{aligned} \quad (1)$$

The N files are mutually independent and uniformly distributed over \mathbb{F}_q^B , that is,

$$\begin{aligned} H(W_1) &= \dots = H(W_N) = B, \\ H(W_1, \dots, W_N) &= H(W_1) + \dots + H(W_N). \end{aligned}$$

Therefore, each subfile or coded subfile is uniformly distributed over $\mathbb{F}_q^{B/L}$. Server $h \in [H]$ stores the h -th coded subfile of each file, i.e.,

$$\bar{W}_{[N],h} := (\bar{W}_{1,h}, \dots, \bar{W}_{N,h}), \quad \forall h \in [H].$$

For notational simplicity, for a vector $\mathbf{a} = (a_1, \dots, a_N)^\top \in \mathbb{F}_q^N$, we denote the scalar (i.e., operations are meant element-wise across files) linear combination of the files or (coded) subfiles for all $l \in [L]$ and $h \in [H]$ as

$$W_{\mathbf{a}} := \sum_{n \in [N]} a_n W_n, \quad (2a)$$

$$W_{\mathbf{a},l} := \sum_{n \in [N]} a_n W_{n,l}, \quad (2b)$$

$$\bar{W}_{\mathbf{a},h} := \sum_{n \in [N]} a_n \bar{W}_{n,h} = \sum_{l \in [L]} g_{l,h} W_{\mathbf{a},l}. \quad (2c)$$

Notice that, $W_{\mathbf{a}}, W_{\mathbf{a},l}, \bar{W}_{\mathbf{a},h}$ are linear in \mathbf{a} , e.g., for any $u, v \in \mathbb{F}_q$ and $\mathbf{a}, \mathbf{b} \in \mathbb{F}_q^N$, $W_{u\mathbf{a}+v\mathbf{b}} = uW_{\mathbf{a}} + vW_{\mathbf{b}}$. Moreover, since $\bar{W}_{n,[H]} := (\bar{W}_{n,1}, \dots, \bar{W}_{n,H})$ is the MDS coded version of $W_{n,[L]} := (W_{n,1}, \dots, W_{n,L})$, $\forall n \in [N]$, by linearity we have that $\bar{W}_{\mathbf{a},[H]} := (\bar{W}_{\mathbf{a},1}, \dots, \bar{W}_{\mathbf{a},H})$ is the MDS coded version of $W_{\mathbf{a},[L]} := (W_{\mathbf{a},1}, \dots, W_{\mathbf{a},L})$, $\forall \mathbf{a} \in \mathbb{F}_q^N$, as in (2c).

The system operates in two phases as follows.

A. Placement Phase

The servers can communicate with each other, and all users can access all servers. To ensure the security condition in (4b), the servers share some randomness V from some finite alphabet \mathcal{V} . Each user $k \in [K]$ generates some random variable P_k from some finite alphabet \mathcal{P}_k and cache some content C_k as a function of P_k, V and the file library $W_{[N]}$. Let the cached content be

$$C_k := \varphi_k(P_k, V, W_{[N]}) \in \mathbb{F}_q^{[MB]}, \quad \forall k \in [K],$$

for some encoding functions $\varphi_k : \mathcal{P}_k \times \mathcal{V} \times \mathbb{F}_q^{NB} \mapsto \mathbb{F}_q^{[MB]}$, $\forall k \in [K]$. The quantity M is referred to as *memory size*. The encoding functions $\varphi_1, \dots, \varphi_K$ are known by the servers, but the randomness P_1, \dots, P_K are kept private by the corresponding users.

B. Delivery Phase

Each user $k \in [K]$ generates a demand $\mathbf{d}_k = (d_{k,1}, \dots, d_{k,N})^\top \in \mathbb{F}_q^N$, meaning it is interested in retrieving the linear combination of the files $W_{\mathbf{d}_k}$. The following random variables are independent

$$\begin{aligned} &H(\mathbf{d}_{[K]}, W_{[N]}, P_{[K]}, V) \\ &= \sum_{k \in [K]} H(\mathbf{d}_k) + \sum_{n \in [N]} H(W_n) + \sum_{k \in [K]} H(P_k) + H(V). \end{aligned}$$

User $k \in [K]$ generates queries $Q_{k,[H]} := (Q_{k,1}, \dots, Q_{k,H})$ as

$$Q_{k,h} := \kappa_{k,h}(\mathbf{d}_k, C_k) \in \mathbb{F}_q^{\ell_{k,h}}, \quad \forall h \in [H],$$

for some query functions $\kappa_{k,h} : \mathbb{F}_q^N \times \mathbb{F}_q^{[MB]} \mapsto \mathbb{F}_q^{\ell_{k,h}}$, where $\ell_{k,h}$ is the length of the query $Q_{k,h}$. If any randomness is needed in the queries, it has to be stored in the cache. Then user $k \in [K]$ sends the query $Q_{k,h}$ to server $h \in [H]$.

Upon receiving the queries from all the users, server $h \in [H]$ creates a signal X_h as

$$X_h := \phi_h(V, Q_{[K],h}, \bar{W}_{[N],h}) \in \mathbb{F}_q^{[R_h B]}, \quad \forall h \in [H],$$

for some encoding function $\phi_h : \mathcal{V} \times \mathbb{F}_q^{\sum_{k \in [K]} \ell_{k,h}} \times \mathbb{F}_q^{\frac{NB}{L}} \mapsto \mathbb{F}_q^{\lfloor R_h B \rfloor}$. The quantity $R_h, h \in [H]$, is referred to as the *load of server h* . The (total) load of the system is defined as

$$R := \sum_{h \in [H]} R_h.$$

An RSP-LFR scheme must satisfy the following conditions for all demands $\mathbf{d}_1, \dots, \mathbf{d}_K \in \mathbb{F}_q^N$.

$$\text{[Robust Correctness]} : H(W_{\mathbf{d}_k} | X_{\mathcal{L}}, \mathbf{d}_k, C_k) = 0, \quad \forall k \in [K], \mathcal{L} \subseteq [H] : |\mathcal{L}| = L, \quad (4a)$$

$$\text{[Security]} : I(W_{[N]}; X_{[H]}) = 0, \quad (4b)$$

$$\text{[User-side Privacy]} : I(\mathbf{d}_{[K] \setminus \mathcal{S}}; C_{\mathcal{S}}, X_{[H]}, \mathbf{d}_{\mathcal{S}}, | W_{[N]}) = 0, \quad \forall \mathcal{S} \subseteq [K] : \mathcal{S} \neq \emptyset, \quad (4c)$$

$$\text{[Server-side Privacy]} : I(\mathbf{d}_{[K]}; Q_{[K],[H]}, \overline{W}_{[N],[H]}, V) = 0, \quad (4d)$$

C. Objective

A memory-load pair $(M, R) \in [1, N] \times \mathbb{R}^+$ is said to be B -achievable if, for any $\epsilon > 0$, there exists a scheme satisfying all the conditions in (4) with memory size less than $M + \epsilon$, load less than $R + \epsilon$ with file-length B . The main objective of this paper is to characterize the optimal load-memory tradeoff of the system, defined as

$$R^*(M) := \inf_{B \in \mathbb{N}^+} \{ R : (M, R) \text{ is } B\text{-achievable} \}. \quad (5)$$

Throughout this paper, we consider the case $N \geq 2$, since demand privacy is impossible for $N = 1$ (i.e., there is only one possible file to be demanded).

For a given scheme, we are also interested in its subpacketization level, which is defined as the number of packets each file has to be partitioned into in order to implement the scheme.

Remark 1 (Implications of the Conditions in (4)): The constraints in (4) imply the following.

- 1) The robust correctness condition in (4a) guarantees that each user can correctly decode its required scalar linear function by receiving any L -subsets of the transmitted signals. Since each user decodes independently, the available subset of signals \mathcal{L} need not to be same across the users.
- 2) The security condition in (4b) guarantees that a wiretapper, who is not a user in the system and observes all the delivery signals, can not obtain any information about the contents of the library files. It was proved in [9, Appendix A] that the conditions in (4b) and (4c) imply

$$I(W_{[N]}, \mathbf{d}_{[K]}; X_{[H]}) = 0,$$

that is, the wiretapper having access to $X_{[H]}$ in fact can not obtain any information on both the files and the demands of the users.

- 3) The user-side privacy condition in (4c) guarantees that any subset of users who exchange their cache contents cannot jointly learn any information on the demands of the other users, regardless of the file realizations.

- 4) The server-side privacy condition in (4d) guarantees that the servers can not obtain any information on the demands of the users, even if all the servers collude by exchanging their stored contents.

Remark 2 (Minimum Memory Size): It was proved in [6] that, in order to guarantee the correctness condition in (4a) and the security condition in (4b) simultaneously, the memory size M has to be no less than one. Thus the load-memory tradeoff is defined for $M \in [1, N]$.

Remark 3 (Comparison With [29]): In the case $L = 1$ and $G = [1, 1, \dots, 1]$, the servers store replicated databases. A scheme to retrieve single files from replicated databases for multiple users was proposed in [29], while guaranteeing server-side privacy. This is different from our setup, even if we remove the user-side privacy and security conditions, since our robust decoding setup in this case imposes that each user can decode from the signal of any single server (i.e., $L = 1$).

Remark 4 (Less Constrained Systems and Naming Convention): For any given RSP-LFR (N, K, L, H) system, the robust correctness condition in (4a) guarantees that the users can correctly decode their demands by receiving the signals from any L servers. In addition to investigating the load-memory tradeoff of the RSP-LFR system, we also discuss less constrained systems where some of the conditions in (4) are relaxed or dropped. In such systems, the optimal load-memory tradeoff can be similarly defined as in (5). In particular, we use $R_{\mathcal{C}}^*(M)$ to denote the optimal load-memory tradeoff of a system with only the constraints listed in the label \mathcal{C} , which can be any of the following:

- L: scalar Linear Function Retrieval (LFR) demands, i.e., the demands $\mathbf{d}_1, \dots, \mathbf{d}_K \in \mathbb{F}_q^N$;
- F: File Retrieval (FR) demands, i.e., the demands $\mathbf{d}_1, \dots, \mathbf{d}_K$ are restricted to $\{\mathbf{e}_1, \dots, \mathbf{e}_K\}$, where $\mathbf{e}_n \in \mathbb{F}_q^N, n \in [N]$, is the vector with the n -th digit being 1 and all the others zero;
- S: the security condition in (4b);
- P: both privacy conditions in (4c) and (4d);
- P_U: the user-side privacy condition in (4c);
- P_S: the server-side privacy condition in (4d);

The convention for the subscript \mathcal{C} is:

- 1) It contains either L or F, but not both, so as to identify the demand type allowed in the system.
- 2) It contains at most one character between P, P_U, P_S, which identifies the privacy condition imposed on the system.
- 3) The tradeoff is defined for $M \in [1, N]$ if it contains S, and for $M \in [0, N]$ otherwise (see Remark 2).

With the above conventions, the value of \mathcal{C} is one from the set

$$\Omega := \{L, LS, LP, LP_S, LP_U, LSP, LSP_S, LSP_U, F, FS, FP, FP_U, FP_S, FSP, FSP_S, FSP_U\}.$$

Notice that, if $\mathcal{C} = LSP$, the system is the novel RSP-LFR setup introduced in this paper, thus, $R_{LSP}^*(M) = R^*(M)$ in (5), defined for all $M \in [1, N]$.

We will also need to discuss the single server system where all the files are stored at the server. The optimal load-memory

tradeoff can be similarly defined for such a system for any constraint implied by $\mathcal{C} \in \Omega$. We will use $\overline{R}_{\mathcal{C}}^*(M)$ to denote the optimal tradeoff in the single server system with constraint identified by $\mathcal{C} \in \Omega$.

III. PDAS AND A TOY EXAMPLE

Our achievable results are based on the notion of PDA [11], originally introduced to reduce the subpacketization in the single-server systems for single file retrieval and without any security or privacy guarantees. In this section, we first review the definition of PDA, and then give an example to highlight the key ideas in the design of our RSP-LFR scheme. The general construction will be discussed in the rest of the paper.

A. Placement Delivery Array

Definition 1 (PDA [11]): For given $K, F \in \mathbb{N}^+$ and $Z, S \in \mathbb{N}$, an $F \times K$ array $\mathbf{A} = [a_{i,j}]$, $i \in [F], j \in [K]$, composed of Z specific symbols “*” in each column and some ordinary symbols $1, \dots, S$, each occurring at least once, is called a (K, F, Z, S) PDA, if, for any two distinct entries $a_{i,j}$ and $a_{i',j'}$, we have $a_{i,j} = a_{i',j'} = s$, for some ordinary symbol $s \in [S]$ only if

- $i \neq i', j \neq j'$, i.e., they lie in distinct rows and distinct columns; and
- $a_{i,j'} = a_{i',j} = *$, i.e., the corresponding 2×2 sub-array formed by rows i, i' and columns j, j' must be of the following form

$$\begin{bmatrix} s & * \\ * & s \end{bmatrix} \text{ or } \begin{bmatrix} * & s \\ s & * \end{bmatrix}.$$

With a given (K, F, Z, S) PDA, it was shown in [11] that there exists an associated coded caching scheme in the single server system without any security or privacy constraint, where the parameter K is the number of users, F is the number of packets each file is split into (i.e., subpacketization), Z is the number of uncoded packets from each file stored at each user, and S is the number of coded multicast signals.

In our model, those implications are used on the subfiles and each single server: the parameter K is the number of users, F is the number of packets each subfile $W_{n,l}$ is split into, Z is the number of uncoded packets from each subfile stored at each user; and S is the number of coded multicast signals sent by each individual server.

B. A Toy RSP-LFR Example From PDAs

We derive here a RSP-LFR scheme associated to the $(K, F, Z, S) = (3, 3, 1, 3)$ PDA

$$\mathbf{A} = \begin{bmatrix} * & 1 & 2 \\ 1 & * & 3 \\ 2 & 3 & * \end{bmatrix} \quad (6)$$

for an $(N, K, L, H) = (4, 3, 2, 3)$ distributed system.

Let the four files be $W_1, W_2, W_3, W_4 \in \mathbb{F}_2^B$ and the $(3, 2)$ generator matrix is given by

$$G = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}. \quad (7)$$

TABLE I
THE CACHED CONTENTS OF USERS[†] ACCORDING TO \mathbf{A} IN (6)

User 1	User 2	User 3
$W_{[4],[2],1}$	$W_{\mathbf{p}_2,[2],1} \oplus V_{[2],1}$	$W_{\mathbf{p}_3,[2],1} \oplus V_{[2],2}$
$W_{\mathbf{p}_1,[2],2} \oplus V_{[2],1}$	$W_{[4],[2],2}$	$W_{\mathbf{p}_3,[2],2} \oplus V_{[2],3}$
$W_{\mathbf{p}_1,[2],3} \oplus V_{[2],2}$	$W_{\mathbf{p}_2,[2],3} \oplus V_{[2],3}$	$W_{[4],[2],3}$

[†] In addition, each user $k \in [3]$ caches \mathbf{p}_k .

That is, each file is split into $L = 2$ subfiles, $W_n = (W_{n,1}, W_{n,2})$, $n \in [4]$ and by (1), the contents stored at the servers are

$$\begin{aligned} \overline{W}_{[4],1} &= W_{[4],1}, \\ \overline{W}_{[4],2} &= W_{[4],2}, \\ \overline{W}_{[4],3} &= W_{[4],1} \oplus W_{[4],2}. \end{aligned}$$

Based on the PDA \mathbf{A} in (6), each subfile $W_{n,l}$ is partitioned into $F = 3$ equal-size packets, $W_{n,l,1}, W_{n,l,2}, W_{n,l,3}$ for all $n \in [4], l \in [2]$. Accordingly, the coded subfile $\overline{W}_{n,h}$ is partitioned into $F = 3$ equal-size packets, $\overline{W}_{n,h,1}, \overline{W}_{n,h,2}, \overline{W}_{n,h,3}$.

Similarly to (2), for any $\mathbf{a} = (a_1, a_2, a_3, a_4)^T \in \mathbb{F}_2^4$, we use the following notation to denote the linear combination of (un)coded packets with coefficient vector \mathbf{a} :

$$\begin{aligned} W_{\mathbf{a},l,i} &:= \bigoplus_{n \in [4]} a_n W_{n,l,i}, \\ \overline{W}_{\mathbf{a},h,i} &:= \bigoplus_{n \in [4]} a_n \overline{W}_{n,h,i} = \bigoplus_{l \in [2]} g_{l,h} W_{\mathbf{a},l,i} \end{aligned}$$

for all $l \in [2], i \in [3], h \in [3]$.

The system operates as follows.

Placement Phase: The servers share $LS = 6$ vectors $\{V_{l,s} : l \in [2], s \in [3]\}$, which are generated independently and uniformly from $\mathbb{F}_2^{B/6}$, where the packets $V_{1,s}, V_{2,s}$ will be associated to the ordinary symbol $s \in [3]$. Each user $k \in [3]$ generates a random vector $\mathbf{p}_k = (p_{k,1}, p_{k,2}, p_{k,3}, p_{k,4})^T \in \mathbb{F}_2^4$. The cache content of the user k is composed of \mathbf{p}_k and the (un)coded packets in the corresponding column in Table I.

The packets $W_{[4],[2],i}$ are associated to the i -th row of \mathbf{A} in (6) and user k is associated to the k -th column of \mathbf{A} . The packets in the i -th row of Table I of user k are created according to the entry $a_{i,k}$ of \mathbf{A} in (6): if $a_{i,k} = *$, user k caches $NL = 8$ uncoded packets $W_{[4],[2],i}$, otherwise it caches $L = 2$ coded packets $W_{\mathbf{p}_k,[2],i} \oplus V_{[2],a_{i,k}}$.

Delivery Phase: Assume that user 1, 2, 3 demands the linear combination $W_{\mathbf{d}_1}, W_{\mathbf{d}_2}$ and $W_{\mathbf{d}_3}$, respectively, where $\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3 \in \mathbb{F}_2^4$. Each user $k \in [3]$ sends $\mathbf{q}_k = \mathbf{p}_k \oplus \mathbf{d}_k$ to all the servers as queries. Upon receiving the query vectors $\mathbf{q}_{[3]}$, each server $h \in [3]$ sends a signal X_h to the users, where X_h is composed of the query vectors $\mathbf{q}_{[3]}$ and $S = 3$ coded packets as in the Table II, which are associated to the ordinary symbols $s = 1, 2, 3$ of \mathbf{A} in (6), respectively, where $(\overline{V}_{1,s}, \overline{V}_{2,s}, \overline{V}_{3,s})$ is the MDS codeword of $(V_{1,s}, V_{2,s})$ with generator matrix G in (7), i.e., for $s \in [3]$,

$$\overline{V}_{1,s} = V_{1,s}, \quad \overline{V}_{2,s} = V_{2,s}, \quad \overline{V}_{3,s} = V_{1,s} \oplus V_{2,s}.$$

Performance: Each user $k \in [3]$ can decode the linear combination $W_{\mathbf{d}_k}$ with signals from any $L = 2$ servers because

TABLE II
THE SIGNALS SENT BY THE SERVERS[†] ACCORDING TO \mathbf{A} IN (6)

s	Server 1	Server 2	Server 3
1	$\bar{V}_{1,1} \oplus \bar{W}_{\mathbf{q}_1,1,2} \oplus \bar{W}_{\mathbf{q}_2,1,1}$	$\bar{V}_{2,1} \oplus \bar{W}_{\mathbf{q}_1,2,2} \oplus \bar{W}_{\mathbf{q}_2,2,1}$	$\bar{V}_{3,1} \oplus \bar{W}_{\mathbf{q}_1,3,2} \oplus \bar{W}_{\mathbf{q}_2,3,1}$
2	$\bar{V}_{1,2} \oplus \bar{W}_{\mathbf{q}_1,1,3} \oplus \bar{W}_{\mathbf{q}_3,1,1}$	$\bar{V}_{2,2} \oplus \bar{W}_{\mathbf{q}_1,2,3} \oplus \bar{W}_{\mathbf{q}_3,2,1}$	$\bar{V}_{3,2} \oplus \bar{W}_{\mathbf{q}_1,3,3} \oplus \bar{W}_{\mathbf{q}_3,3,1}$
3	$\bar{V}_{1,3} \oplus \bar{W}_{\mathbf{q}_2,1,3} \oplus \bar{W}_{\mathbf{q}_3,1,2}$	$\bar{V}_{2,3} \oplus \bar{W}_{\mathbf{q}_2,2,3} \oplus \bar{W}_{\mathbf{q}_3,2,2}$	$\bar{V}_{3,3} \oplus \bar{W}_{\mathbf{q}_2,3,3} \oplus \bar{W}_{\mathbf{q}_3,3,2}$

[†] In addition, each server $h \in [3]$ transmits the query vectors $\mathbf{q}_{[3]}$.

TABLE III
THE SIGNALS A USER CAN DECODE FROM THE TRANSMISSION BY THE SERVERS[†] ACCORDING TO \mathbf{A} IN (6)

s	Subfile 1	Subfile 2
1	$V_{1,1} \oplus W_{\mathbf{q}_1,1,2} \oplus W_{\mathbf{q}_2,1,1}$	$V_{2,1} \oplus W_{\mathbf{q}_1,2,2} \oplus W_{\mathbf{q}_2,2,1}$
2	$V_{1,2} \oplus W_{\mathbf{q}_1,1,3} \oplus W_{\mathbf{q}_3,1,1}$	$V_{2,2} \oplus W_{\mathbf{q}_1,2,3} \oplus W_{\mathbf{q}_3,2,1}$
3	$V_{1,3} \oplus W_{\mathbf{q}_2,1,3} \oplus W_{\mathbf{q}_3,1,2}$	$V_{2,3} \oplus W_{\mathbf{q}_2,2,3} \oplus W_{\mathbf{q}_3,2,2}$

[†] In addition, each server $h \in [3]$ transmits the query vectors $\mathbf{q}_{[3]}$.

user k can decode $W_{\mathbf{d}_{k,[2],k}}$ since it has cached all the uncoded packets $W_{[4],[2],k}$ from Table I. For the other packets, we note:

- For each $s \in [3]$, the signals associated to s over the servers form an MDS codeword with generator matrix G , whose original packets are coded packets within each subfile as shown in Table III. By the property of MDS codes, each user can decode the signals in Table III by receiving signals from any $L = 2$ of the servers.
- Upon obtaining the signals in Table III, each user $k \in [3]$ can proceed with the decoding process for each subfile $l \in [2]$ as in [9]. Let us take $s = 1$ for subfile $l = 1$ as an example. As $a_{1,2} = a_{2,1} = 1$, user 1 can decode $W_{\mathbf{d}_{1,1,2}}$ and user 2 can decode $W_{\mathbf{d}_{2,1,1}}$ from the signal $V_{1,1} \oplus W_{\mathbf{q}_1,1,2} \oplus W_{\mathbf{q}_2,1,1}$, i.e.,

$$W_{\mathbf{d}_{1,1,2}} = (V_{1,1} \oplus W_{\mathbf{q}_1,1,2} \oplus W_{\mathbf{q}_2,1,1}) \quad (8a)$$

$$\oplus (V_{1,1} \oplus W_{\mathbf{p}_1,1,2}) \quad (8b)$$

$$\oplus W_{\mathbf{q}_2,1,1}, \quad (8c)$$

thus, user 1 can decode $W_{\mathbf{d}_{1,1,2}}$ since the signals in (8b) are cached by user 1, and the signal in (8c) can be computed from the cached uncoded packets $W_{[4],1,1}$ and the vector \mathbf{q}_2 . Similarly, user 2 can decode the packet $W_{\mathbf{d}_{2,1,1}}$ by computing

$$W_{\mathbf{d}_{2,1,1}} = (V_{1,1} \oplus W_{\mathbf{q}_1,1,2} \oplus W_{\mathbf{q}_2,1,1})$$

$$\oplus (V_{1,1} \oplus W_{\mathbf{p}_2,1,1})$$

$$\oplus W_{\mathbf{q}_1,1,2}.$$

One can verify that each user $k \in [3]$ can decode all the remaining packets $W_{\mathbf{d}_{k,[2],[3]\setminus\{k\}}}$ from its stored contents, the signals in Table III and the query vectors $\mathbf{q}_{[3]}$.

This concludes the proof of correct robust decoding. Privacy and security are guaranteed since each signal is accompanied by a key of random and uniformly distributed bits.

In term of memory-load performance, recall that each packet is of size $\frac{B}{6}$ bits. Each user caches 12 packets and 1 vectors

in \mathbb{F}_2^4 , whose length does not scale with B . Thus the needed memory is $M = 12 \times \frac{1}{6} = 2$ files. Each of the 3 server sends 3 packets and 3 vectors in \mathbb{F}_2^4 , thus the achieved load is $R = 3 \times 3 \times \frac{1}{6} = \frac{3}{2}$ files. Hence, the scheme achieves the memory-load pair $(M, R) = (2, \frac{3}{2})$.

IV. MAIN RESULTS

A. PDA Based RSP-LFR Schemes

With any given PDA, we will construct an associated RSP-LFR scheme. The following theorem summarizes the performance of PDA based SP-LFR scheme, which will be proved by presenting and analyzing the construction in Section V.

Theorem 1: For any (N, K, L, H) system and a given (K, F, Z, S) PDA \mathbf{A} , there exists an associated RSP-LFR scheme that achieves the memory-load pair

$$(M_{\mathbf{A}}, R_{\mathbf{A}}) = \left(1 + \frac{Z}{F}(N-1), \frac{H}{L} \cdot \frac{S}{F}\right). \quad (10)$$

with subpacketization LF .

Remark 5 (Comparison With Single-Server Systems):

With the procedure described in Section V, we can easily obtain RSP-LFR schemes from existing PDA constructions, such as those in [11]–[15]. If $H = L = 1$, the system degrades to a single-server shared-link system, where all the files are stored at the server [2]. In [10], a key superposition scheme was proposed to guarantee the correctness, security, and user privacy conditions simultaneously based on any (K, F, Z, S) PDA \mathbf{A} for single-server systems. The scheme in [10] achieves the memory-load pair in (10) with $H/L = 1$. In other words, the RSP-LFR scheme with PDA \mathbf{A} achieves the same memory size as in the single server case but the load is scaled by a factor $\frac{H}{L}$. In the case $H = L$, each user needs to retrieve information from all the servers, and the total load is the same as that from a single server case (i.e., $H = L = 1$). Moreover, this indicates that, in addition to guaranteeing correctness, security, and user-side privacy conditions, the server-side privacy condition does not increase the load-memory tradeoff in the non-robust multi-server case with $H = L$.

B. Optimality of MAN-PDA Based RSP-LFR Schemes

The following PDA describing the MAN scheme in [2] is important, and will be referred to as MAN-PDA in the following.

Definition 2 (MAN-PDA): For any integer $j \in [0 : K]$, define the set $\Omega_j \triangleq \{\mathcal{T} \subseteq [K] : |\mathcal{T}| = j\}$. Fix any integer

$t \in [0 : K]$, denote the set $\Omega_t = \{\mathcal{T}_i : i \in [\binom{K}{t}]\}$. Also, choose an arbitrary bijective function κ_{t+1} from Ω_{t+1} to the set $[\binom{K}{t+1}]$. Then, define the array $\mathbf{A}_t = [a_{i,j}]$ as

$$a_{i,j} \triangleq \begin{cases} *, & \text{if } j \in \mathcal{T}_i \\ \kappa_{t+1}(\{j\} \cup \mathcal{T}_i), & \text{if } j \notin \mathcal{T}_i. \end{cases} \quad (11)$$

It was proved in [11] that \mathbf{A}_t from (11) in Definition 2 is a $(K, \binom{K}{t}, \binom{K-1}{t-1}, \binom{K}{t+1})$ PDA.

Example 1 (A MAN-PDA): Consider $K = 4$, $t = 2$, let $\mathcal{T}_1 = \{1, 2\}$, $\mathcal{T}_2 = \{1, 3\}$, $\mathcal{T}_3 = \{1, 4\}$, $\mathcal{T}_4 = \{2, 3\}$, $\mathcal{T}_5 = \{2, 4\}$ and $\mathcal{T}_6 = \{3, 4\}$. Let κ_3 be the lexicographic order of a subset of size 3 in Ω_3 , e.g., $\kappa_3(\{1, 2, 3\}) = 1$, $\kappa_3(\{1, 2, 4\}) = 2$ and $\kappa_3(\{1, 3, 4\}) = 3$ and $\kappa_3(\{2, 3, 4\}) = 4$. The corresponding $(4, 6, 3, 4)$ PDA is given by

$$\mathbf{A}_2 = \begin{bmatrix} * & * & 1 & 2 \\ * & 1 & * & 3 \\ * & 2 & 3 & * \\ 1 & * & * & 4 \\ 2 & * & 4 & * \\ 3 & 4 & * & * \end{bmatrix}.$$

The following theorem summarizes the performance of MAN-PDA and its optimality. The proof is presented in Section VI-A.

Theorem 2: Let $R(M)$ be the lower convex envelope of the following points

$$(M_t, R_t) = \left(1 + \frac{t(N-1)}{K}, \frac{H(K-t)}{L(t+1)}\right), \quad (12)$$

where $t \in [0 : K]$, then $R(M)$ is achievable in an (N, K, L, H) RSP-LFR system, where the point (M_t, R_t) can be achieved with subpacketization $L \binom{K}{t}$. Moreover, $R(M)$ and the optimal communication load $R^*(M)$ satisfies

1) $N \geq K$, for all $M \in [1, N)$,

$$\frac{R(M)}{R^*(M)} \leq \begin{cases} 1, & \text{if } K = 1 \\ 2, & \text{if } N = K = 2 \\ 6.02652, & \text{if } N = K \geq 3 \\ 5.0221, & \text{if } N = K + 1 \\ 4.01768, & \text{if } N \geq K + 2. \end{cases}$$

2) $N < K$, for all $M \in [2, N)$,

$$\frac{R(M)}{R^*(M)} < 8.$$

Remark 6 (On Unbounded Regime): In the regime $N < K, 1 \leq M < 2$ the gap is unbounded. From our proof, $\frac{R(M)}{R^*(M)}$ is upper bounded by $\frac{\bar{R}_{\text{LSP}_U}(M)}{\bar{R}_{\text{LSP}_U}^*(M)}$, where $\bar{R}_{\text{LSP}_U}(M)$ is the tradeoff achieved by the key superposition scheme in the single server system where the security and user-side privacy conditions are imposed [10], and $\bar{R}_{\text{LSP}_U}^*(M)$ is the corresponding optimal tradeoff. The gap result in Theorem 2 thus follows from the bound for $\frac{\bar{R}_{\text{LSP}_U}(M)}{\bar{R}_{\text{LSP}_U}^*(M)}$ in [10], where the same regime is open. The main problem in this regime for the single server model is that, if security keys are used [6], [10], for the point $M = 1$ the best known achievable load is K , while

the best known converse is N . Thus, it seems that the larger load when $K > N$ is mainly caused by the security condition; closing the gap in small memory regime is an open problem in the S-FR setup [6]. When new converse and gap will be obtained for this regime in the single server case, the same gap will apply to our RSP-LFR system. In fact, it was observed in [10] that, $\frac{\bar{R}_{\text{LSP}_U}(M)}{\bar{R}_{\text{LSP}_U}^*(M)}$ is upper bounded by the constant 17 for the regime $N > K$ and $1 + \frac{(N-K)(N-1)}{NK} \leq M \leq 2$, which was a gap obtained for S-FR setup [6]. Thus, the gap remains unbounded only for the regime $K > N, 1 \leq M \leq 1 + \frac{(N-K)(N-1)}{NK}$.

The following theorem implies that, with the given procedure of deriving RSP-LFR scheme in Section V, the memory-load pairs $\{(M_t, R_t) : t \in [0 : K]\}$ achieved by the MAN-PDAs are Pareto-optimal among all PDA based RSP-LFR schemes. Moreover, the MAN-PDAs have the smallest subpacketization among all PDA based RSP-LFR schemes achieving these points. The proof is deferred to Section VI-B.

Theorem 3: Given a (K, F, Z, S) PDA, if the associated RSP-LFR scheme achieves a memory-load pair (M, R) , then necessarily

$$R \geq \frac{HK(N-M)}{L(N-1+K(M-1))} = \frac{H(K-x)}{L(x+1)} \Big|_{x=K \frac{M-1}{N-1}}. \quad (13)$$

In particular, the memory-load pairs $\{(M_t, R_t) : t \in [0 : K]\}$ satisfy (13) with equality. Moreover, if $M = M_t$ and $R = R_t$ for some $t \in [0 : K]$, then the subpacketization is at least $L \binom{K}{t}$.

Remark 7 (Subpacketizations): By the procedure described in Section V, we can easily obtain RSP-LFR schemes from existing PDA constructions, such as those in [11]–[15]. It was shown in [10] that the new PDA construction based scheme in [11] achieves a slightly larger load than MAN-PDA for the same memory size, while reducing the subpacketization by a factor that increases exponentially with K . Thus, PDAs in [11] sacrifice some load for an exponential reduction in subpacketization.

C. Improved Load-Memory Tradeoffs for Less Constrained Systems

Obviously, the load-memory tradeoff $R(M)$ in Theorem 2 is achievable for any less constrained system described in Remark 4. In this subsection, we present improved achievable results for the following three less constrained systems. The details are presented in Section VII.

1) *Robust Private Linear Function Retrieval (RP-LFR) System* ($\mathcal{C} = \text{LP}$): In an (N, K, L, H) RP-LFR system, the correctness condition (4a) and the privacy conditions (4c)–(4d) must be guaranteed for all LFR demands.

Theorem 4: For an (N, K, L, H) RP-LFR system, let $R_{\text{LP}}(M)$ be the lower convex envelope of the point $(0, \frac{HN}{L})$ and the following points

$$(M_t^{\text{LP}}, R_t^{\text{LP}}) := \left(1 + \frac{t(N-1)}{K}, \frac{H\left(\binom{K}{t+1} - \binom{K-\min\{K, N\}}{t+1}\right)}{L \binom{K}{t}}\right), \quad (14)$$

where $t \in [0 : K]$. Then, $R_{\text{LP}}(M)$ is achievable, and it satisfies

$$\frac{R_{\text{LP}}(M)}{R_{\text{LP}}^*(M)} \leq 6.3707, \quad \forall M \in [0, N].$$

2) *Robust Private File Retrieval (RP-FR) System* ($\mathcal{C} = \text{FP}$): In an (N, K, L, H) RP-FR system, the correctness condition (4a) and the privacy conditions (4c)–(4d) must be guaranteed for all FR demands.

Theorem 5: For an (N, K, L, H) RP-FR system, let $R_{\text{FP}}(M)$ be the lower convex envelope of the point $(0, \frac{HN}{L})$ and the following points

$$(M_t^{\text{FP}}, R_t^{\text{FP}}) := \left(1 + \frac{t(N-1)}{K}, \frac{H \left(\binom{K}{t+1} - \binom{K-\min\{K, N-1\}}{t+1} \right)}{L \binom{K}{t}} \right),$$

where $t \in [0 : K]$. Then, $R^{\text{RP-F}}(M)$ is achievable, and it satisfies

$$\frac{R_{\text{FP}}(M)}{R_{\text{FP}}^*(M)} \leq 5.4606, \quad \forall M \in [0, N].$$

3) *Robust Linear Function Retrieval (R-LFR) System* ($\mathcal{C} = \text{L}$): In an (N, K, L, H) R-LFR system, only the correctness condition (4a) must be guaranteed for all LFR demands.

Theorem 6: For an (N, K, L, H) R-LFR system, let $R_{\text{L}}(M)$ be the lower convex envelope of the following points

$$(M_t^{\text{L}}, R_t^{\text{L}}) := \left(\frac{tN}{K}, \frac{H \left(\binom{K}{t+1} - \binom{K-\min\{K, N\}}{t+1} \right)}{L \binom{K}{t}} \right),$$

where $t \in [0 : K]$. Then, $R_{\text{L}}(M)$ is achievable and it satisfies

$$\frac{R_{\text{L}}(M)}{R_{\text{L}}^*(M)} \leq 2.00884, \quad \forall M \in [0, N].$$

Remark 8 (Less Constrained Systems): Notice that if $R_{\mathcal{C}}(M)$ is achievable for the constraint \mathcal{C} , then $R_{\mathcal{C}}(M)$ is achievable for all constraints that are less restrictive than \mathcal{C} . In particular, with Theorem 2, the tradeoff $R(M)$ is achievable for all $\mathcal{C} \in \Omega$. Moreover, with Theorems 4–6, the tradeoff

- 1) $R_{\text{LP}}(M)$ is achievable for any $\mathcal{C} \in \{\text{LP}, \text{LP}_S, \text{LP}_U\}$;
- 2) $R_{\text{FP}}(M)$ is achievable for any $\mathcal{C} \in \{\text{FP}, \text{FP}_S, \text{FP}_U\}$;
- 3) $R_{\text{L}}(M)$ is achievable for any $\mathcal{C} \in \{\text{L}, \text{F}\}$.

Moreover, from the proofs in Section VII, it is clear that the subpacketization for $(M_t^{\mathcal{C}}, R_t^{\mathcal{C}})$ is $L \binom{K}{t}$ for all $t \in [0 : K]$ and $\mathcal{C} \in \{\text{LP}, \text{FP}, \text{L}\}$ (and thus also for their less constrained systems).

V. PROOF OF THEOREM 1

In this section, we derive a RSP-LFR scheme for an (N, K, L, H) system from any given (K, F, Z, S) PDA $\mathbf{A} = [a_{i,j}]_{F \times K}$. Based on \mathbf{A} , each subfile $W_{n,l}$ ($n \in [N], l \in [F]$) is partitioned into F equal-size packets, denoted by $W_{n,l,1}, \dots, W_{n,l,F}$, where each packet $W_{n,l,i} \in \mathbb{F}_q^{B/(LF)}$. The packets with index i , i.e., $W_{[N],[L],i}$, are associated to the i -th row of \mathbf{A} . According to (1), each coded subfile $\bar{W}_{n,h}$

($n \in [N], h \in [H]$) is composed of F coded packets, denoted by $\bar{W}_{n,h,1}, \dots, \bar{W}_{n,h,F}$, where

$$\bar{W}_{n,h,i} = \sum_{l \in [L]} g_{l,h} W_{n,l,i}, \quad \forall i \in [F].$$

That is, the coded contents stored at server h are

$$\bar{W}_{[N],h,[F]}, \quad \forall h \in [H].$$

We use the following notations similarly to (2) for any $\mathbf{a} = (a_1, \dots, a_N)^\top \in \mathbb{F}_q^N$ to denote the linear combination of (un)coded packets:

$$\begin{aligned} rClW_{\mathbf{a},l,i} &= \sum_{n \in [N]} a_n W_{n,l,i}, \quad \forall l \in [L], i \in [F]. \\ \bar{W}_{\mathbf{a},h,i} &= \sum_{n \in [N]} a_n \bar{W}_{n,h,i}, \quad \forall h \in [H], i \in [F]. \end{aligned}$$

Notice that $(\bar{W}_{\mathbf{a},1,i}, \dots, \bar{W}_{\mathbf{a},H,i})$ is the MDS codeword of $(W_{\mathbf{a},1,i}, \dots, W_{\mathbf{a},L,i})$, i.e.,

$$\begin{aligned} &(\bar{W}_{\mathbf{a},1,i}, \dots, \bar{W}_{\mathbf{a},H,i}) \\ &= \left(\sum_{l \in [L]} g_{l,1} W_{\mathbf{a},l,i}, \dots, \sum_{l \in [L]} g_{l,H} W_{\mathbf{a},l,i} \right). \end{aligned} \quad (15)$$

Moreover, $W_{\mathbf{a},l,i}, \bar{W}_{\mathbf{a},h,i}$ are linear in \mathbf{a} .

A. Placement Phase

the servers share the random variables

$$V = \{V_{l,s} : l \in [L], s \in [S]\}, \quad (16)$$

which are SL vectors independently and uniformly distributed over $\mathbb{F}_q^{B/(FL)}$. Each user $k \in [K]$ locally generates a random vector \mathbf{p}_k uniformly over \mathbb{F}_q^N , and constructs its local cache C_k as

$$C_k = \{\mathbf{p}_k\} \quad (17a)$$

$$\cup \{W_{n,l,i} : n \in [N], l \in [L], i \in [F], a_{i,k} = *\} \quad (17b)$$

$$\cup \{W_{\mathbf{p}_k,l,i} + V_{l,a_{i,k}} : l \in [L], i \in [F], a_{i,k} \neq *\}. \quad (17c)$$

B. Delivery Phase

Assume that user $k \in [K]$ demands $W_{\mathbf{d}_k}$, for some $\mathbf{d}_k \in \mathbb{F}_q^N$. Then user $k \in [K]$ sends query $\mathbf{q}_k = \mathbf{d}_k + \mathbf{p}_k$ to all the servers, i.e., the queries $Q_{k,[H]}$ are constructed as

$$Q_{k,h} = \mathbf{q}_k = \mathbf{d}_k + \mathbf{p}_k, \quad \forall h \in [H]. \quad (18)$$

For each $s \in [S]$, consider the MDS coded version of $(V_{1,s}, \dots, V_{L,s})$ with the generator matrix G , i.e.,

$$(\bar{V}_{1,s}, \dots, \bar{V}_{H,s}) = \left(\sum_{l \in [L]} g_{l,1} V_{l,s}, \dots, \sum_{l \in [L]} g_{l,H} V_{l,s} \right). \quad (19)$$

Upon receiving the queries $Q_{[K],h} = \mathbf{q}_{[K]}$, each server $h \in [H]$ sends the signal

$$X_h = (\mathbf{q}_{[K]}, \bar{\mathbf{Y}}_{h,[S]}) \quad (20)$$

to the users, where for each $s \in [S]$, $\bar{Y}_{h,s}$ is

$$\bar{Y}_{h,s} = \bar{V}_{h,s} + \sum_{\substack{(u,v) \in [F] \times [K] \\ a_{u,v} = s}} \bar{W}_{\mathbf{q}_{v,h,u}}. \quad (21)$$

C. Robust Correctness

We need to show that for each user $k \in [K]$, with any $\mathcal{L} \subseteq [K]$ such that $|\mathcal{L}| = L$, user k can decode its demanded scalar linear function $W_{\mathbf{d}_k}$, i.e., all the packets $W_{\mathbf{d}_k, [L], [F]}$.

For each $i \in [F]$ such that $a_{i,k} = *$, by (17b), user $k \in [K]$ has stored all the packets $W_{[N], [L], i}$, thus it can directly compute the packets $W_{\mathbf{d}_k, l, i}$ for each $l \in [L]$.

Now, consider any $i \in [F]$ such that $a_{i,k} \neq *$. Let $s \triangleq a_{i,k}$, notice that by (15) and (19), $(\bar{Y}_{1,s}, \dots, \bar{Y}_{H,s})$ is the MDS coded version of information coded packets $(Y_{1,s}, \dots, Y_{L,s})$ with generator matrix G , where

$$Y_{l,s} := V_{l,s} + \sum_{\substack{(u,v) \in [F] \times [K] \\ a_{u,v} = s}} W_{\mathbf{q}_{v,l,u}}, \quad \forall l \in [L]. \quad (22)$$

By the property of MDS codes, each user can decode all the L coded packets in (22) with signals from any subset of L servers for each $s \in [S]$. Since $a_{i,k} = s$, for each $l \in [L]$, the signal $Y_{l,s}$ in (22) can be written as

$$\begin{aligned} Y_{l,s} &= V_{l,s} + W_{\mathbf{q}_k, l, i} + \sum_{\substack{(u,v) \in [F] \times [K] \\ a_{u,v} = s, (u,v) \neq (i,k)}} W_{\mathbf{q}_{v,l,u}} \\ &\stackrel{(a)}{=} W_{\mathbf{d}_k, l, i} + (V_{l, a_{i,k}} + W_{\mathbf{p}_k, l, i}) \\ &\quad + \sum_{\substack{(u,v) \in [F] \times [K] \\ a_{u,v} = s = a_{i,k}, (u,v) \neq (i,k)}} W_{\mathbf{q}_{v,l,u}}, \end{aligned}$$

where (a) follows from $\mathbf{q}_k = \mathbf{p}_k + \mathbf{d}_k$. Therefore, user $k \in [K]$ can decode $W_{\mathbf{d}_k, l, i}$ from the signal $Y_{l,s}$ by canceling the remaining terms since

- 1) the coded packet $V_{l, a_{i,k}} + W_{\mathbf{p}_k, l, i}$ is cached by user k by (17c);
- 2) for each $(u,v) \in [F] \times [K]$ such that $a_{u,v} = s$ and $(u,v) \neq (i,k)$, since $a_{i,k} = a_{u,v} = s$, by the definition of PDA, $i \neq u, v \neq k$ and $a_{i,v} = a_{u,k} = *$. Thus, user $k \in [K]$ stores all the packets $W_{[N], [L], u}$. Hence, user k can compute $W_{\mathbf{q}_{v,l,u}}$ for each $l \in [L]$.

Remark 9 (On the Robust Decoding): From the above decoding process, user $k \in [K]$ can decode its demanded linear function if for any $i \in [F]$ such that $a_{i,k} \neq *$, user k can receive any L of the coded signals $\bar{Y}_{1, a_{i,k}}, \dots, \bar{Y}_{H, a_{i,k}}$. This is less restrictive than the assumptions in our setup (i.e., each user can obtain a fixed subset of signals $X_{\mathcal{L}}$), since (i) it allows the available subset \mathcal{L} to vary over different transmission $s \in [S]$; (ii) it only needs to decode packets over the signals associated to s such that, $a_{i,k} = s$ for some $i \in [F]$, which indicates that for $s \in [S] \setminus \{a_{i,k} : i \in [F]\}$, the availability of the signals $\bar{Y}_{[H], s}$ does not affect the decodability of user k .

D. Security

We have

$$I(W_{[N]}; X_{[H]}) \quad (23a)$$

$$= I(W_{[N]}; \mathbf{q}_{[K]}, \bar{Y}_{[H], [S]}) \quad (23b)$$

$$= I(W_{[N]}; \mathbf{q}_{[K]}, Y_{[L], [S]}) \quad (23c)$$

$$= I(W_{[N]}; \mathbf{q}_{[K]}) + I(W_{[N]}; Y_{[L], [S]} | \mathbf{q}_{[K]}) \quad (23d)$$

$$= 0, \quad (23e)$$

where: (23c) holds since $(\bar{Y}_{1,s}, \dots, \bar{Y}_{H,s})$ is the MDS coded version of $(Y_{1,s}, \dots, Y_{L,s})$ for each $s \in [S]$, and hence they determine each other; and (23e) follows since (a) the vectors $\mathbf{q}_{[K]} = \mathbf{d}_{[K]} + \mathbf{p}_{[K]}$ are independent of $W_{[N]}$, and (b) $Y_{[L], [S]}$ are independent of $(W_{[N]}, \mathbf{q}_{[K]})$ because the random variables $V_{[L], [S]}$ are independently and uniformly distributed.

E. User-Side Privacy in (4c)

We have

$$I(\mathbf{d}_{[K] \setminus S}; C_S, X_{[H]}, \mathbf{d}_S | W_{[N]}) \quad (24a)$$

$$= I(\mathbf{d}_{[K] \setminus S}; C_S, \mathbf{q}_{[K]}, \bar{Y}_{[H], [S]}, \mathbf{d}_S | W_{[N]}) \quad (24b)$$

$$= I(\mathbf{d}_{[K] \setminus S}; C_S, \mathbf{q}_{[K]}, Y_{[L], [S]}, \mathbf{d}_S | W_{[N]}) \quad (24c)$$

$$= 0, \quad (24d)$$

where: (24c) follows since $\bar{Y}_{[H], [S]}$ and $Y_{[L], [S]}$ determine each other due to the fact that $\bar{Y}_{[H], s}$ is the MDS coded version of $Y_{[L], s}$ for each $s \in [S]$; and (24d) follows since $\mathbf{d}_{[K] \setminus S} = \mathbf{q}_{[K] \setminus S} - \mathbf{p}_{[K] \setminus S}$ is independent of $(C_S, W_{[N]}, \mathbf{q}_{[K]}, \mathbf{d}_S, Y_{[L], [H]})$ since $\mathbf{p}_{[K] \setminus S}$ are independently and uniformly distributed.

F. Server-Side Privacy in (4d)

We have

$$I(\mathbf{d}_{[K]}; Q_{[K], [H]}, \bar{W}_{[N], [H]}, V) \quad (25a)$$

$$= I(\mathbf{d}_{[K]}; \mathbf{q}_{[K]}, W_{[N]}, V) \quad (25b)$$

$$= I(\mathbf{d}_{[K]}; W_{[N]}, V) + I(\mathbf{d}_{[K]}; \mathbf{q}_{[K]} | W_{[N]}, V) \quad (25c)$$

$$= 0, \quad (25d)$$

where: (25b) follows from (18) and the fact $\bar{W}_{[N], [H]}$ and $W_{[N]}$ determines each other; and (25d) holds because (a) $\mathbf{d}_{[K]}$ is independent of $(W_{[N]}, V)$; (b) $\mathbf{q}_{[K]} = \mathbf{p}_{[K]} + \mathbf{d}_{[K]}$ are independent of $(\mathbf{d}_{[K]}, W_{[N]}, V)$ since the vectors $\mathbf{p}_{[K]}$ are independent random variables uniformly distributed.

G. Performance

By construction, each subfile is split into F equal-size packets, each of length $\frac{B}{LF}$ symbols, thus the subpacketization is LF . For each user $k \in [K]$, by the cached content in (17), for each $i \in [F]$ such that $a_{i,k} = *$, there are LN associated packets cached by the user, one from each file (see (17b)). For each $i \in [F]$ such that $a_{i,k} \neq *$, there are L associated coded packet cached at the user (see (17c)). In addition, the \mathbf{p}_k in (17a) can be stored with N symbols. Recall that, each

column of a (K, F, Z, S) PDA has Z “*”s and $F - Z$ ordinary symbols, thus, the needed cache size is

$$\begin{aligned} M_{\mathbf{A}} &= \inf_{B \in \mathbb{N}^+} \frac{1}{B} \left((Z \cdot LN + (F - Z)L) \frac{B}{LF} + N \right) \\ &= \frac{F + Z(N - 1)}{F}. \end{aligned}$$

By (20), each server $h \in [H]$ sends S coded packets $Y_{h,[S]}$, each of size $\frac{B}{LF}$ symbols, and the coefficient vectors $\mathbf{q}_{[K]}$ can be sent in KN symbols, thus the achieved load is

$$R_{\mathbf{A}} = \inf_{B \in \mathbb{N}^+} \frac{1}{B} \left(HS \frac{B}{LF} + H \cdot KN \right) = \frac{HS}{LF}.$$

Remark 10 (Novality of This Work): Compared to our previous work on SP-LFR systems in [10], this work extends the key superposition scheme in [10] to distributed systems. The prefetched contents are packets of original files, while the multicast signals are formed by the MDS coded packets, as indicated by (17) and (21), respectively. This inserts the property of robust decoding in the distributed setup as mentioned in Remark 9. As distributed systems are widely used in practice, due to their advantages such as less storage space at each server, robustness against server failures and so on, the proposed scheme indicates that the techniques such as multicast coding, key superpositions from the single server system can be integrated with the robustness provided by MDS code and applied to distributed systems.

VI. MAN-PDA AND ITS OPTIMALITY

A. MAN-PDA: Performance and Gap Results (Proof of Theorem 2)

The achievability of the point (M_t, R_t) directly follows from Theorem 1 and the $(K, \binom{K}{t}, \binom{K-1}{t-1}, \binom{K}{t+1})$ MAN-PDA \mathbf{A}_t in Definition 2. Moreover, the lower convex envelope of the points in (12) can be achieved by memory-sharing technique [2].

For the gap result, we derive the following lemma for any $\mathcal{C} \in \Omega$.

Lemma 1: For any $\mathcal{C} \in \Omega$, for any feasible² M ,

$$R_{\mathcal{C}}^*(M) \geq \frac{H}{L} \cdot \bar{R}_{\mathcal{C}}^*(M),$$

Proof: For a (N, K, L, H) system with the constraint \mathcal{C} , for any feasible design of caches $Z_{[K]}$ and signals $X_{[H]}$ satisfying the constraint \mathcal{C} , for any $\mathcal{L} \subseteq [H]$, the contents $Z_{[K]}$ and signal $X \triangleq X_{\mathcal{L}}$ are a feasible scheme for the single server system with the same constraint \mathcal{C} . Thus,

$$\frac{H(X_{\mathcal{L}})}{B} \geq \bar{R}_{\mathcal{C}}^*(M), \quad \forall \mathcal{L} \subseteq [K], |\mathcal{L}| = L.$$

Therefore,

$$R_{\mathcal{C}}^*(M) \geq \frac{1}{B} \sum_{h \in [H]} H(X_h) \quad (26a)$$

$$= \frac{H}{B} \cdot \frac{1}{H} \sum_{h \in [H]} H(X_h) \quad (26b)$$

²If \mathcal{C} contains S, $M \in [1, N]$; else $M \in [0, N]$ (see Remark 2).

$$\geq \frac{H}{B} \cdot \frac{1}{\binom{H}{L}} \sum_{\mathcal{L} \subseteq [H], |\mathcal{L}|=L} \frac{H(X_{\mathcal{L}})}{L} \quad (26c)$$

$$= H \cdot \frac{1}{\binom{H}{L}} \sum_{\mathcal{L} \subseteq [H], |\mathcal{L}|=L} \frac{R_{\mathcal{C}}^*(M)}{L} \quad (26d)$$

$$\geq \frac{H}{L} \cdot \bar{R}_{\mathcal{C}}^*(M), \quad (26e)$$

where (26c) follows from Han's inequality [34]. ■

Let $\bar{R}_{\text{LSP}_U}(M)$ be the lower convex envelope of the following points: for each $t \in [0 : K]$,

$$(\bar{M}_t, \bar{R}_t) = \left(1 + \frac{t(N-1)}{K}, \frac{K-t}{t+1} \right), \quad (27)$$

Notice that $\bar{R}_{\text{LSP}_U}(M)$ is achievable by the key superposition scheme in [10] for the single server system with constraint LSP_U . Comparing (12) with (27), we see that $R(M) = \frac{H}{L} \cdot \bar{R}_{\text{LSP}_U}(M)$ (see also Remark 5), hence by Lemma 1, for all $M \in [1, N]$,

$$\frac{R(M)}{R^*(M)} \leq \frac{\bar{R}_{\text{LSP}_U}(M)}{\bar{R}_{\text{LSP}}^*(M)} \stackrel{(a)}{\leq} \frac{\bar{R}_{\text{LSP}_U}(M)}{\bar{R}_{\text{LSP}_U}^*(M)}. \quad (28)$$

where (a) follows from the fact $\bar{R}_{\text{LSP}}^*(M) \geq \bar{R}_{\text{LSP}_U}(M)$, since the constraint LSP is stronger than the constraint LSP_U . Thus, the claimed multiplicative gap result directly follows from (28) and the bound for $\frac{\bar{R}_{\text{LSP}_U}(M)}{\bar{R}_{\text{LSP}_U}^*(M)}$ in [10, Theorem 3].

B. MAN-PDA: Optimality Within PDA Based RSP-LFR Schemes (Proof of Theorem 3)

Consider a single server network with constraint LSP_U as in [10]. For any (K, F, Z, S) , the scheme proposed in [10] from PDA \mathbf{A} achieves the memory-load pair $(\bar{M}_{\mathbf{A}}, \bar{R}_{\mathbf{A}}) = (1 + \frac{Z(N-1)}{F}, \frac{S}{F})$. The following conclusion was proved in [10].

Lemma 2 (From [10, Theorem 2]): Given a (K, F, Z, S) PDA \mathbf{A} , if the associated scheme for the single server system with constraint LSP_U achieves a memory-load pair $(\bar{M}_{\mathbf{A}}, \bar{R}_{\mathbf{A}})$, then necessarily

$$\bar{R}_{\mathbf{A}} \geq \frac{K(N - \bar{M}_{\mathbf{A}})}{N - 1 + K(\bar{M}_{\mathbf{A}} - 1)}. \quad (29)$$

In particular, the memory-load pairs $\{(\bar{M}_t, \bar{R}_t) : t \in [0 : K]\}$ given in (27) satisfy (29) with equality. Moreover, if $\bar{M}_{\mathbf{A}} = \bar{M}_t$ and $\bar{R}_{\mathbf{A}} = \bar{R}_t$ for some $t \in [0 : K]$, then $F \geq \binom{K}{t}$.

Now consider a (K, F, Z, S) PDA \mathbf{A} . Assume that the associated RSP-LFR scheme achieves the memory-load pair $(M, R) = (M_{\mathbf{A}}, R_{\mathbf{A}})$, then

$$R = R_{\mathbf{A}} = \frac{H}{L} \cdot \bar{R}_{\mathbf{A}} \quad (30a)$$

$$\geq \frac{H}{L} \cdot \frac{K(N - \bar{M}_{\mathbf{A}})}{N - 1 + K(\bar{M}_{\mathbf{A}} - 1)} \quad (30b)$$

$$= \frac{HK(N - M)}{L(N - 1 + K(M - 1))} \quad (30c)$$

$$= \frac{H(K - x)}{L(1 + x)} \Big|_{x=K \frac{M-1}{N-1}}, \quad (30d)$$

where: (30a) follows from Remark 5; (30b) follows from (29); and (30c) follows from the fact $M = M_{\mathbf{A}} = \overline{M}_{\mathbf{A}}$ by Remark 5. Therefore, we proved (13).

The fact that memory pairs $\{(M_t, R_t) : t \in [0 : K]\}$ satisfy (13) with equality can be verified trivially. Moreover, if $M = M_{\mathbf{A}} = M_t$ and $R = R_{\mathbf{A}} = R_t$, then $\overline{M}_{\mathbf{A}} = \overline{M}_t$ and $\overline{R}_{\mathbf{A}} = \overline{R}_t$, by the facts $M_t = \overline{M}_t$, $R_t = \frac{H}{L} \cdot \overline{R}_t$ and Remark 5. Therefore, by Lemma 2, it must hold that $F \geq \binom{K}{t}$. Thus, by Theorem 1, the subpacketization of the RSP-LFR scheme is at least $L \binom{K}{t}$.

VII. IMPROVED LOAD-MEMORY TRADEOFFS IN LESS CONSTRAINED SYSTEMS

The basic idea for improving the load-memory tradeoff in less constrained systems is that in the case the security condition (4b) is not imposed (i.e., the constraint \mathcal{C} does not contain S), some redundant signals may be removed when $N \leq K$ as in [4], [5]. Notice that in such less constrained systems, $R_{\mathcal{C}}^*(M)$ is defined on $M \in [0, N]$.

Consider a fixed MAN-PDA \mathbf{A}_t in (11), where $F = \binom{K}{t}$ and $S = \binom{K}{t+1}$. Notice that each row of \mathbf{A}_t is associated to a subset of size t , i.e., for any given $\mathbf{a} \in \mathbb{F}_q^N$ and $l \in [L]$ or $h \in [H]$, each linear combination of files $W_{\mathbf{a},l,u}$ or $\overline{W}_{\mathbf{a},h,u}$ is associated to the subset $\mathcal{T}_u \subseteq [K]$. For notational simplicity, in this section, for each $u \in \binom{K}{t}$, denote

$$W_{\mathbf{a},l,\mathcal{T}_u} := W_{\mathbf{a},l,u}, \quad \overline{W}_{\mathbf{a},h,\mathcal{T}_u} := \overline{W}_{\mathbf{a},h,u}. \quad (31)$$

Moreover, each signal $Y_{l,s}$ or $\overline{Y}_{h,s}$ is associated to a subset $\mathcal{J} \subseteq [K]$ of size $t+1$, i.e., the subset \mathcal{J} such that $s = \kappa_{t+1}(\mathcal{J})$. Denote

$$Y_{l,\mathcal{J}} := Y_{l,\kappa_{t+1}(\mathcal{J})}, \quad \overline{Y}_{h,\mathcal{J}} := \overline{Y}_{h,\kappa_{t+1}(\mathcal{J})}. \quad (32)$$

In RP-LFR, RP-FR and R-LFR systems, the security condition (4b) is not imposed. Thus, the security keys can be dropped, i.e., instead of generating the random variables in (16) we set

$$V_{l,s} = \mathbf{0}, \quad \forall l \in [L], s \in [S]. \quad (33)$$

Therefore, with notations as in (31) and (32), by (21) and (22), we have

$$Y_{l,\mathcal{J}} = \sum_{j \in \mathcal{J}} W_{\mathbf{a}_j,l,\mathcal{J} \setminus \{j\}}, \quad \overline{Y}_{h,\mathcal{J}} = \sum_{j \in \mathcal{J}} \overline{W}_{\mathbf{a}_j,h,\mathcal{J} \setminus \{j\}},$$

where $(\overline{Y}_{1,\mathcal{J}}, \dots, \overline{Y}_{H,\mathcal{J}})$ is the MDS coded version of $(Y_{1,\mathcal{J}}, \dots, Y_{L,\mathcal{J}})$ with generator matrix G .

A. Improved Tradeoff in RP-LFR

System (Proof of Theorem 4)

In RP-LFR system, the robust correctness, user-side and server-side privacy conditions are guaranteed for all LFR demands. Notice that, the point $(0, \frac{HN}{L})$ can be achieved by trivially transmitting the whole coded subfiles $\overline{W}_{[N],h}$ to the users for any server $h \in [H]$. The point $(M_K^{\text{LP}}, R_K^{\text{LP}}) = (N, 0)$ can be achieved by trivially storing all the N files at each user. In the following, we describe the scheme achieving the point

$(M_t^{\text{LP}}, R_t^{\text{LP}})$ in (14) for each fixed $t \in [0 : K-1]$. The lower convex envelope of those points can be achieved by memory-sharing technique.

1) *Placement Phase*: The cached contents of the users are generated as in (17) according to \mathbf{A}_t , i.e., with notations as in (31), user $k \in [K]$ caches

$$C_k = \{W_{n,l,\mathcal{T}} : n \in [N], l \in [L], \mathcal{T} \subseteq [K], |\mathcal{T}| = t, k \in \mathcal{T}\} \quad (34a)$$

$$\cup \{W_{\mathbf{p}_k,l,\mathcal{T}} : l \in [L], \mathcal{T} \subseteq [K], |\mathcal{T}| = t, k \notin \mathcal{T}\} \quad (34b)$$

$$\cup \{\mathbf{p}_k\}. \quad (34c)$$

2) *Delivery Phase*: The queries $\mathbf{q}_{[K]}$ are generated as in (18). Let $\mathcal{I} \subseteq [K]$ be a subset such that the vectors $\mathbf{q}_{\mathcal{I}}$ form a maximum linear independent vector group of the vectors $\mathbf{q}_{[K]}$. Each server $h \in [H]$ sends

$$X_h^{\text{LP}} = (\mathbf{q}_{[K]}, \overline{\mathbf{Y}}_h(\mathcal{I})), \quad (35)$$

where

$$\overline{\mathbf{Y}}_h(\mathcal{I}) \triangleq \{\overline{Y}_{h,\mathcal{J}} : \mathcal{J} \subseteq [K], |\mathcal{J}| = t+1, \mathcal{J} \cap \mathcal{I} \neq \emptyset\}. \quad (36)$$

3) *Robust Correctness*: For any fixed $\mathcal{J} \subseteq [K]$ of size $t+1$, $(\overline{Y}_{1,\mathcal{J}}, \dots, \overline{Y}_{H,\mathcal{J}})$ is the MDS coded version of $(Y_{1,\mathcal{J}}, \dots, Y_{L,\mathcal{J}})$ with generator matrix G . Thus with signals from any L servers, each user can decode

$$\{Y_{l,\mathcal{J}} : l \in [L], \mathcal{J} \subseteq [K], |\mathcal{J}| = t+1, \mathcal{J} \cap \mathcal{I} \neq \emptyset\}.$$

Moreover, for each fixed $l \in [L]$, by the results in [5], the signals $\{Y_{l,\mathcal{J}}\}_{\mathcal{J} \subseteq [K], |\mathcal{J}|=t+1}$ can be decoded from the signals $\{Y_{l,\mathcal{J}}\}_{\mathcal{J} \subseteq [K], |\mathcal{J}|=t+1, \mathcal{J} \cap \mathcal{I} \neq \emptyset}$. As a result, each user $k \in [K]$ can decode

$$\begin{aligned} \{Y_{l,\mathcal{J}} : l \in [L], \mathcal{J} \subseteq [K], |\mathcal{J}| = t+1\} \\ = \{Y_{l,s} : l \in [L], s \in [S]\}, \end{aligned}$$

i.e., all the signals in (22). By continue with the same arguments following (22), each user can correctly decode its demanded linear combination of the files.

4) *User/Server-Side Privacy*: The proof that the scheme guarantees the server-side and user-side privacy conditions follow the same line of reasoning as in (24) and (25), respectively.

5) *Performance*: By (34), each user stores $NL \binom{K-1}{t-1} + L \binom{K-1}{t}$ packets, each of size $\frac{B}{L \binom{K}{t}}$, and a vector $\mathbf{p}_k \in \mathbb{F}_q^N$ of length N . The needed memory size is given by

$$\begin{aligned} M_t^{\text{LP}} &= \inf_{B \in \mathbb{N}^+} \frac{1}{B} \left(\frac{B(NL \binom{K-1}{t-1} + L \binom{K-1}{t})}{L \binom{K}{t}} + N \right) \\ &= 1 + \frac{t(N-1)}{K}. \end{aligned}$$

Let $\text{rank}_q(\mathbf{q}_{[K]})$ be the rank of vectors $\mathbf{q}_{[K]}$, i.e., the cardinality of \mathcal{I} . By (35) and (36), each server sends $\binom{K}{t+1} - \binom{K - \text{rank}_q(\mathbf{q}_{[K]})}{t+1}$ packets, and K vectors of length N . Notice that the worst case is $\text{rank}_q(\mathbf{q}_{[K]}) = \min\{N, K\}$, therefore,

the load is given by

$$\begin{aligned} R_t^{\text{LP}} &= \inf_{B \in \mathbb{N}^+} \frac{1}{B} \left(\frac{HB \left(\binom{K}{t+1} - \binom{K-\min\{K,N\}}{t+1} \right)}{L \binom{K}{t}} + NK \right) \\ &= \frac{H \left(\binom{K}{t+1} - \binom{K-\min\{K,N\}}{t+1} \right)}{L \binom{K}{t}}. \end{aligned}$$

Gap Result: Let $\bar{R}_{\text{LP}_U}(M)$ be the load-memory tradeoff achieved by the scheme in [9] in the single server case, where user-side privacy is guaranteed for all LFR demands, which is given by the lower convex envelope of the point $(0, N)$ and the following points

$$\begin{aligned} (\bar{M}_t^{\text{LP}_U}, \bar{R}_t^{\text{LP}_U}) &= \left(1 + \frac{t(N-1)}{K}, \frac{\binom{K}{t+1} - \binom{K-\min\{K,N\}}{t+1}}{\binom{K}{t}} \right), \end{aligned}$$

where $t \in [0 : K]$. Notice that, for the corner points with $M = 0$ and $M \in \{M_t^{\text{LP}_U} : t \in [0 : K]\}$, it always hold

$$R_{\text{LP}}(M) = \frac{H}{L} \cdot \bar{R}_{\text{LP}_U}(M). \quad (37)$$

Since the corner points coincide on M , (37) hold for all $M \in [0, N]$. Moreover,

$$\frac{R_{\text{LP}}(M)}{\bar{R}_{\text{LP}}^*(M)} \stackrel{(a)}{\leq} \frac{\bar{R}_{\text{LP}_U}(M)}{\bar{R}_{\text{LP}}^*(M)} \stackrel{(b)}{\leq} \frac{\bar{R}_{\text{LP}_U}(M)}{\bar{R}_{\text{LP}_U}^*(M)}, \quad (38)$$

where: (a) follows from Lemma 1 and (37); and (b) follows from the fact $\bar{R}_{\text{LP}}^*(M) \geq \bar{R}_{\text{LP}_U}^*(M)$, since the constraint LP is stronger than the constraint LP_U . Then the gap result directly follows from bound for $\frac{\bar{R}_{\text{LP}_U}(M)}{\bar{R}_{\text{LP}_U}^*(M)}$ in [9, Theorem 6].

B. Improvement in RP-FR System (Proof of Theorem 5)

In RP-FR system, the robust correctness, user-side and server-side privacy conditions are guaranteed for all FR demands. The proof of Theorem 5 follows similarly to the proof of Theorem 4 in Section VII-A, with the following distinctions.

1) *Placement Phase:* Instead of generating $\mathbf{p}_1, \dots, \mathbf{p}_K$ uniformly from \mathbb{F}_q^N , we let $\mathbf{p}_1, \dots, \mathbf{p}_K$ generated uniformly from $\{(x_1, \dots, x_N)^\top \in \mathbb{F}_q^N : \sum_{n \in [N]} x_n = q-1\}$.

2) *Performance:* Since the queries $\mathbf{q}_1, \dots, \mathbf{q}_K$ are generated as in (18) and the demands $\mathbf{d}_1, \dots, \mathbf{d}_K \in \{\mathbf{e}_1, \dots, \mathbf{e}_N\}$, the queries are uniformly distributed over the $N-1$ dimensional subspace $\{(x_1, \dots, x_N)^\top \in \mathbb{F}_q^N : \sum_{n \in [N]} x_n = 0\}$. Thus, in the worst case, $\text{rank}_q(\mathbf{q}_{[K]}) = \min\{K, N-1\}$. As a result, the achieved memory-load pair $(M_t^{\text{FP}}, R_t^{\text{FP}})$ is given by

$$\begin{aligned} (M_t^{\text{FP}}, R_t^{\text{FP}}) &= \left(1 + \frac{t(N-1)}{K}, \frac{H \left(\binom{K}{t+1} - \binom{K-\min\{K,N-1\}}{t+1} \right)}{L \binom{K}{t}} \right). \end{aligned}$$

3) *Gap Result:* Let $\bar{R}_{\text{FP}_U}(M)$ be the lower convex envelope of the point $(0, N)$ and points $\{(\bar{M}_t^{\text{FP}_U}, \bar{R}_t^{\text{FP}_U}) : t \in [0 : K]\}$, where

$$\begin{aligned} (\bar{M}_t^{\text{FP}_U}, \bar{R}_t^{\text{FP}_U}) &= \left(1 + \frac{t(N-1)}{K}, \frac{\binom{K}{t+1} - \binom{K-\min\{K,N-1\}}{t+1}}{\binom{K}{t}} \right), \end{aligned}$$

which is proved to be achievable in the single server case for all FR demands in [9, Theorem 1]. Following the same line of reasoning as to obtain (38), we have $R_{\text{FP}}(M) = \frac{H}{L} \cdot \bar{R}_{\text{FP}_U}(M)$ for all $M \in [0, N]$, and

$$\frac{R_{\text{FP}}(M)}{\bar{R}_{\text{FP}}^*(M)} \leq \frac{\bar{R}_{\text{FP}_U}(M)}{\bar{R}_{\text{FP}}^*(M)} \leq \frac{\bar{R}_{\text{FP}_U}(M)}{\bar{R}_{\text{FP}_U}^*(M)}.$$

Then gap result directly follows from the upper bound for $\frac{\bar{R}_{\text{LP}_U}(M)}{\bar{R}_{\text{LP}_U}^*(M)}$ in [9, Theorem 5].

C. Improvement in R-LFR System (Proof of Theorem 6)

In the R-LFR system, only the robust correctness condition must be guaranteed for all LFR demands. As a result, in addition to dropping the security keys (see (33)), the privacy keys can also be dropped, i.e., set to zero. In particular, the stored contents in (34b) and (34c) can be dropped, i.e., set to zero. The correctness can be easily verified by setting $\mathbf{p}_1 = \dots = \mathbf{p}_K = \mathbf{0}$ and following the same line of reasoning as in Section VII-A. The distinctions are in performance and gap results.

1) *Performance:* In the modified scheme for R-LFR system, only the contents in (34a) are stored. The delivered signals are the same as in (35). Thus, the achieved memory-load pair is given by

$$(M_t^{\text{L}}, R_t^{\text{L}}) = \left(\frac{tN}{K}, \frac{H \left(\binom{K}{t+1} - \binom{K-\min\{K,N\}}{t+1} \right)}{L \binom{K}{t}} \right),$$

where $t \in [0 : K]$. The lower convex envelope of those points can be achieved by memory-sharing.

2) *Gap Result:* Let $\bar{R}_{\text{F}}(M)$ be the lower convex envelope of the points $\{(\bar{M}_t^{\text{F}}, \bar{R}_t^{\text{F}}) : t \in [0 : K]\}$ where

$$(\bar{M}_t^{\text{F}}, \bar{R}_t^{\text{F}}) = \left(\frac{tN}{K}, \frac{\binom{K}{t+1} - \binom{K-\min\{K,N\}}{t+1}}{\binom{K}{t}} \right),$$

which is proved to be achievable in the single server case for all FR demands in [4]. Following the same line of reasoning as to obtain (38), we have $R_{\text{L}}(M) = \frac{H}{L} \cdot \bar{R}_{\text{F}}(M)$ for all $M \in [0, N]$, and

$$\frac{R_{\text{L}}(M)}{\bar{R}_{\text{L}}^*(M)} \leq \frac{\bar{R}_{\text{F}}(M)}{\bar{R}_{\text{L}}^*(M)} \leq \frac{\bar{R}_{\text{F}}(M)}{\bar{R}_{\text{F}}^*(M)}.$$

Then the gap result follows from the upper bound for $\frac{\bar{R}_{\text{F}}(M)}{\bar{R}_{\text{F}}^*(M)}$ in [36, Theorem 1].

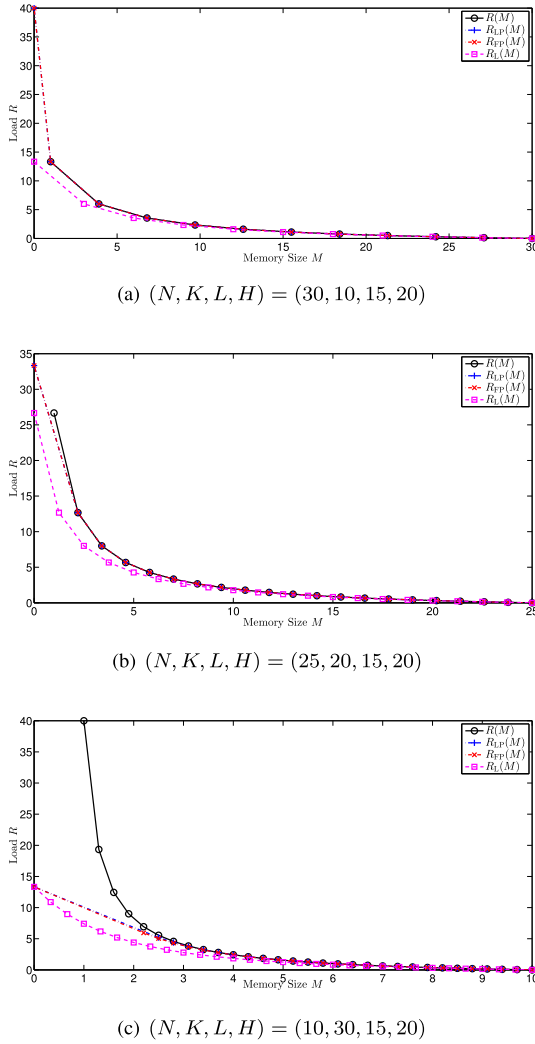


Fig. 2. Load-memory tradeoffs for robust systems (a) $N \geq \frac{K+1+\sqrt{3K^2+1}}{2}$, (b) $K < N < \frac{K+1+\sqrt{3K^2+1}}{2}$; (c) $N \leq K$.

VIII. NUMERICAL RESULTS

In Fig. 2, we plot the achievable memory-load tradeoff under different constraints (Theorems 2, 4, 5 and 6) for the three regimes:

- $N \geq \frac{K+1+\sqrt{3K^2+1}}{2}$, see Fig. (2a);
- $K < N < \frac{K+1+\sqrt{3K^2+1}}{2}$, see Fig. (2b); and
- $N \leq K$, see Fig. (2c).

We choose parameters $(N, K, L, H) = (30, 10, 15, 20), (25, 20, 15, 20), (10, 30, 15, 20)$, respectively. From the figures, we observe:

1) For $N \geq \frac{K+1+\sqrt{3K^2+1}}{2}$ (Fig. (2a)), the MAN-PDA based scheme in the RSP-LFR system achieves the same tradeoff as that in the RP-LFR and RP-FR systems on the interval $M \in [1, N]$. This is because:

- there is no redundant signals to be removed in RP-LFR or RP-FR;
- the privacy keys and security keys are stored in the superposition form;
- the lower convex envelope of $(0, \frac{HN}{L})$ and $\{(M_t, R_t) : t \in [0 : K]\}$ are formed by connecting $(0, \frac{HN}{L})$ and

$(M_0, R_0), (M_1, R_1), \dots, (M_K, R_K)$ sequentially. This can be verified by letting the slope of the line connecting $(0, \frac{HN}{L})$ and (M_0, R_0) be no larger than the slope of connecting (M_0, R_0) and (M_1, R_1) , i.e.,

$$\frac{R_0 - HN/L}{M_0 - 0} \leq \frac{R_1 - R_0}{M_1 - M_0}, \quad (39)$$

which indicates that N should satisfy $N \geq \frac{K+1+\sqrt{3K^2+1}}{2}$.

The improved tradeoff in R-LFR system is due to the saved memory for keys for the regime $M \in [1, N]$, and there is no need to guarantee privacy by sending all coded files at $M = 0$ (i.e., the point $(0, K)$ is achievable in R-LFR system).

2) For $K < N < \frac{K+1+\sqrt{3K^2+1}}{2}$ (Fig. (2b)), similar phenomena are observed as in the case $N \geq \frac{K+1+\sqrt{3K^2+1}}{2}$, except that now there is slightly improvement in RP-LFR and RP-FR systems over the RSP-LFR system in the interval $M \in [1, 1 + \frac{N-1}{K}]$. This improvement comes from taking the lower convex envelope with the additional point $(0, \frac{HN}{L})$ (observe that (39) does not hold). Notice that for the case $N > K$ (Fig. (2a) and (2b)), all the tradeoffs are proved to be within a constant multiplicative gap of the optimal tradeoff in their respective setups.

3) For the case $N \leq K$ (Fig. (2c)), the tradeoff in RP-LFR and RP-FR systems significantly smaller than that in the LSP-LFR system for small M regime, because:

- The trivial point $(M, R) = (0, \frac{HN}{L})$ can be achieved, and thus memory-sharing the other points with this point increases the performance.
- For $M \in \{M_t : t \in [0 : K - N]\}$, some redundant signals are removed in RP-LFR and RP-FR, similarly to [4], [5].

In this case, due to the use of security keys in the RSP-LFR system, the counterpart of redundant signals in RP-LFR and RP-FR system can not be obtained from the counterpart of the transmitted signals. Notice that, the tradeoff in RP-FR is slightly better than that in the RP-LFR system, since the number of removed redundant signals in RP-FR system is $\binom{K-N+1}{t+1}$, which is larger than that in the RP-LFR system $\binom{K-N}{t+1}$. The improvement in the R-LFR system over RP-LFR/RP-FR systems comes from the saved memory size for privacy keys.

IX. CONCLUSION

A PDA-based key superposition RSP-LFR scheme is proposed for MDS distributed storage systems that simultaneously guarantees content security against a wiretapper having access to the delivery signals and demand privacy against both servers and colluding users. The load-memory tradeoff turns out to be the single-server one scaled by the inverse of the rate of the MDS code in order to guarantee robustness against link/server failures. The performance of MAN-PDA-based RSP-LFR scheme is shown to be to within a multiplicative gap of at most eight from optimal in all regimes, except for small memory regime with less files than users. Moreover, in three less restrictive systems without the security constraint (i.e., RP-LFR, RP-FR, and R-LFR systems), some redundant signals

can be removed to further improve the load-memory tradeoff, which are proved to be within a constant multiplicative gap of the optimal tradeoff in their respective setups.

REFERENCES

- [1] Q. Yan and D. Tuninetti, "Robust and secure cache-aided private linear function retrieval from coded servers," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Melbourne, VIC, Australia, Jul. 2021, pp. 1–23.
- [2] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.
- [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, Jan. 2020.
- [4] 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.
- [5] K. Wan, H. Sun, M. Ji, D. Tuninetti, and G. Caire, "On the optimal load-memory tradeoff of cache-aided scalar linear function retrieval," *IEEE Trans. Inf. Theory*, vol. 67, no. 6, pp. 4001–4018, Jun. 2021.
- [6] 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.
- [7] K. Wan and G. Caire, "On coded caching with private demands," *IEEE Trans. Inf. Theory*, vol. 67, no. 1, pp. 358–372, Jan. 2021.
- [8] S. Kamath, "Demand private coded caching," 2019, *arXiv:1909.03324*.
- [9] Q. Yan and D. Tuninetti, "Fundamental limits of caching for demand privacy against colluding users," *IEEE J. Sel. Area. Inf. Theory*, vol. 2, no. 1, pp. 192–207, Mar. 2021.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] M. Cheng, J. Jiang, X. Tang, and Q. Yan, "Some variant of known coded caching schemes with good performance," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1370–1377, Mar. 2020.
- [16] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," in *Proc. 36th Annu. Symp. Found. Comput. Sci.*, 1995, pp. 41–50.
- [17] H. Sun and S. A. Jafar, "The capacity of private information retrieval," *IEEE Trans. Inf. Theory*, vol. 63, no. 7, pp. 4075–4088, Jul. 2017.
- [18] H. Sun and S. A. Jafar, "The capacity of private computation," *IEEE Trans. Inf. Theory*, vol. 65, no. 6, pp. 3880–3897, Jun. 2019.
- [19] H. Sun and S. A. Jafar, "The capacity of private information retrieval with colluding databases," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Washington, DC, USA, Dec. 2016, pp. 941–946.
- [20] R. Tandon, "The capacity of cache aided private information retrieval," in *Proc. 55th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Monticello, IL, USA, Oct. 2017, pp. 1078–1082.
- [21] Y.-P. Wei, K. Banawan, and S. Ulukus, "Cache-aided private information retrieval with partially known uncoded prefetching: Fundamental limits," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1126–1139, Jun. 2018.
- [22] Y.-P. Wei, K. Banawan, and S. Ulukus, "Fundamental limits of cache-aided private information retrieval with unknown and uncoded prefetching," *IEEE Trans. Inf. Theory*, vol. 65, no. 5, pp. 3215–3232, May 2019.
- [23] Y.-P. Wei, K. Banawan, and S. Ulukus, "The capacity of private information retrieval with partially known private side information," *IEEE Trans. Inf. Theory*, vol. 65, no. 12, pp. 8222–8231, Dec. 2019.
- [24] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network codes for distributed storage," *Proc. IEEE*, vol. 99, no. 3, pp. 476–489, Mar. 2011.
- [25] K. Banawan and S. Ulukus, "The capacity of private information retrieval from coded databases," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1945–1956, Mar. 2017.
- [26] J. Zhu, Q. Yan, C. Qi, and X. Tang, "A new capacity-achieving private information retrieval scheme with (almost) optimal file length for coded servers," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1248–1260, 2020.
- [27] R. Zhou, C. Tian, T. Liu, and H. Sun, "Capacity-achieving private information retrieval codes from mds-coded databases with minimum message size," *IEEE Trans. Inf. Theory*, vol. 66, no. 8, pp. 4904–4916, Aug. 2020.
- [28] R. Tajeddine and S. E. Rouayheb, "Robust private information retrieval on coded data," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Aachen, Germany, Jun. 2017, pp. 1903–1907.
- [29] X. Zhang, K. Wan, H. Sun, M. Ji, and G. Caire, "On the fundamental limits of cache-aided multiuser private information retrieval," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 5828–5842, Sep. 2021.
- [30] X. Zhang, K. Wan, H. Sun, M. Ji, and G. Caire, "A new design of cache-aided multiuser private information retrieval with uncoded prefetching," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Melbourne, VIC, Australia, Jul. 2021, pp. 1–7.
- [31] V. R. Aravind, P. K. Sarvepalli, and A. Thangaraj, "Subpacketization in coded caching with demand privacy," in *Proc. Nat. Conf. Commun. (NCC)*, Kharagpur, India, Feb. 2020, pp. 1–6.
- [32] S. Kamath, J. Ravi, and B. K. Dey, "Demand-private coded caching and the exact tradeoff for $N = K = 2$," in *Proc. Nat. Conf. Commun. (NCC)*, Kharagpur, India, Feb. 2020, pp. 1–6.
- [33] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.
- [34] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2012.
- [35] S. Fujishige, *Submodular Functions and Optimization*. Amsterdam, The Netherlands: Elsevier, 2005.
- [36] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Characterizing the rate-memory tradeoff in cache networks within a factor of 2," *IEEE Trans. Inf. Theory*, vol. 65, no. 1, pp. 647–663, Jan. 2019.



Qifa Yan (Member, IEEE) received the B.S. degree in mathematics and applied mathematics from Shanxi University, Taiyuan, China, in 2010, and the Ph.D. degree in communication and information systems from the School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China, in 2017. From November 2017 to October 2019, he was a Joint Post-Doctoral Researcher with Télécom Paris, Institut Polytechnique de Paris, and CentraleSupélec, Paris-Saclay University, France. From January 2020 to January 2021, he was a Post-Doctoral Research Fellow with the Department of Electrical and Computer Engineering, University of Illinois Chicago, USA. He is currently an Assistant Professor with the School of Information Science and Technology, Southwest Jiaotong University. His research interests include caching networks, distributed computing, other fields related to wireless networks, information theory, and coding theory.



Daniela Tuninetti (Fellow, IEEE) received the Ph.D. degree in electrical engineering from ENST/Télécom ParisTech, Paris, France, with work done at the Eurecom Institute in Sophia Antipolis, France, in 2002. In 2005, she joined the Department of Electrical and Computer Engineering, University of Illinois Chicago (UIC), where she is currently a Professor. She was a Post-Doctoral Research Associate at the School of Communication and Computer Science, Swiss Federal Institute of Technology in Lausanne (EPFL), Lausanne, Switzerland, from 2002 to 2004. Her research interests include ultimate performance limits of wireless interference networks (with special emphasis on cognition and user cooperation), content-type coding, cache-aided systems, and distributed private coded computing, with security or privacy constraints. She was a recipient of the Best Paper Award at the European Wireless Conference in 2002, NSF CAREER Award in 2007, and the University of Illinois Scholar in 2015. She is currently a Distinguished Lecturer of the Information Theory Society. She was the Editor-in-Chief of the IEEE Information Theory Society Newsletter from 2006 to 2008. She was an editor of IEEE COMMUNICATION LETTERS from 2006 to 2009, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS from 2011 to 2014, IEEE TRANSACTIONS ON INFORMATION THEORY from 2014 to 2017, and IEEE TRANSACTIONS ON COMMUNICATIONS since 2021.