Robust and Secure Cache-aided Private Linear Function Retrieval from Coded Servers

Qifa Yan and Daniela Tuninetti
University of Illinois Chicago, Chicago, IL 60607, USA, Email:{qifa, danielat}@uic.edu

***Abstract*—This paper investigates the ultimate performance limits of Linear Function Retrieval (LFR) by cache-aided users from distributed coded servers. Each user aims to retrieve a linear function of the files of a library, which are Maximum Distance Separable (MDS) coded and stored at multiple servers. The system needs to guarantee robust decoding in the sense that each user must decode its demanded function with signals from any subset of servers whose cardinality exceeds a threshold. In addition, the following conditions must be met: (a) the content of the library must be kept secure from a wiretapper who obtains all the signals sent by 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. A scheme that uses the superposition of security and privacy keys is proposed to meet all those conditions. The achieved load-memory tradeoff is the same as that achieved in single-server case scaled by the inverse of the MDS code rate used to encode the files, and the same optimality guarantees as in single-server setup are obtained.**

I. Introduction

Coded caching, introduced by Maddah-Ali and Niesen (MAN) [1], is a technique to reduce the communication load by leveraging the multicast opportunities created by caches at the users. The model consists of a single-server, multiple users, and two phases. In the *placement phase*, the users' caches are filled 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 a shared link. It turns out that for a system with N files and K users, the MAN scheme achieves the optimal load-memory tradeoff among all uncoded placement schemes when $N \geq K$ [2], and after removing some redundant transmissions, also for $N < K$ [3]. Recently, it was showed that allowing the users to demand arbitrary linear combinations of the files does not increase the load compared to single file retrieval [4].

In practical systems, content security and demand privacy are both critical aspects. In [5], the content of the files must be protected against a wiretapper who obtains the delivery phase transmissions. We investigated demand-privacy against colluding users in [6] for both single file retrieval and linear function retrieval, where 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 [5] and [6] is that users cache, in addition to the content as in the MAN scheme [1], also some *security keys* or *privacy keys* for the MAN uncached part of the files; this is done in a structured way so that each user can retrieve all the multicast

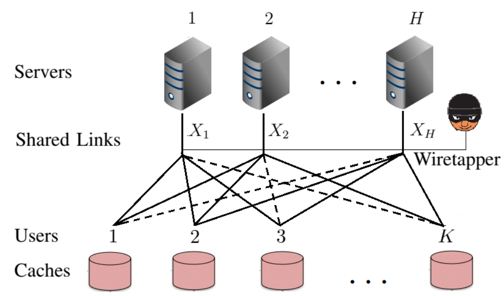

Fig. 1: System model

messages needed for correct decoding. We investigated the content Secure and demand Private Linear Function Retrieval (SP-LFR) problem in [7], For linear function retrieval [4] with security [5] and user-privacy constraints [6]. We designed a *key superposition* scheme to guarantee the security and privacy conditions simultaneously by superposing (i.e., sum together) the security keys and privacy keys. It was showed that the load-memory tradeoff is the same as in the setup with only content security guarantees.

Since node failures and erasures arise naturally in any storage system, redundancy should be introduced [8]. A common erasure coding technique is to use Maximum Distance Separable (MDS) codes. An (H, L) MDS code encodes L packets into H packets, with the property that upon obtaining any L (out of H) coded packets one can recover the L information packets. This motivates us to investigate cache-aided LFR from a distributed storage system [9]–[11], as in Fig. 1. The model consists of H servers, where each file is stored at the servers in the form of (H, L) MDS coded version. Each server is connected to the users via a dedicated shared link, but may not reach all of the users. The coding scheme needs to guarantee that each user can retrieval an arbitrary linear function of the files from the signals of arbitrary L servers. The security [5] and user-privacy [6] conditions are also imposed. In addition, the users' demands must kept private against all the servers, even if they exchange their available information, which we refer to as server-privacy. We propose a scheme that builds on the key superposition idea from [7]. In particular, key superposition is used on the MDS coded packets in the delivery phase. Interestingly, both the achieved load and converse are increased by a factor $\frac{H}{L}$ compared to the the single server case [7], and thus the same optimality guarantees of the single-server case continue to hold in the multi-server case.

II. SYSTEM MODEL

Let N, K, L, H be positive integers satisfying $L \leq H$. The (N, K, L, H) system illustrated in Fig. 1 consists of H servers (denoted by $1, \ldots, H$), where each server is connected to K users (denoted by $1, \ldots, K$) via a dedicated shared-link. A file library of N files (denoted by $W_1, \ldots, W_N \in \mathbb{F}_q^B$) 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 $W_{n,1}, \ldots, W_{n,L} \in \mathbb{F}_q^{B/L}$ and is encoded into H coded subfiles $\overline{W}_{n,1}, \ldots, \overline{W}_{n,H}$ with a given (H, L) MDS code with generator matrix

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

that is, the coded subfiles are given by

$$\overline{W}_{n,h} = \sum_{l \in [L]} g_{l,h} W_{n,l}, \quad \forall\, h \in [H], n \in [N]. \tag{2}$$

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

$$H(W_1) = \ldots = H(W_N) = B, \tag{3a}$$

$$H(W_1, \ldots, W_N) = H(W_1) + \ldots + H(W_N). \tag{3b}$$

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., $\overline{W}_{[N],h} := (\overline{W}_{1,h}, \ldots, \overline{W}_{N,h})$.

For notational simplicity, for any given vector $\mathbf{a} = (a_1, \ldots, a_N)^\top \in \mathbb{F}_q^N$, we denote the linear combination of the files or (coded) subfiles for all $l \in [L]$ and $h \in [H]$ as

$$W_{\mathbf{a}} := \sum_{n \in [N]} a_n W_n, \quad W_{\mathbf{a},l} := \sum_{n \in [N]} a_n W_{n,l}, \tag{4a}$$

$$\overline{W}_{\mathbf{a},h} := \sum_{n \in [N]} a_n \overline{W}_{n,h} = \sum_{l \in [L]} g_{l,h} W_{\mathbf{a},l}. \tag{4b}$$

Notice that, $W_{\mathbf{a}}, W_{\mathbf{a},l}, \overline{W}_{\mathbf{a},h}$ 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 $\overline{W}_{n,[H]} := (\overline{W}_{n,1}, \ldots, \overline{W}_{n,H})$ is the MDS coded version of $W_{n,[L]} := (W_{n,1}, \ldots, W_{n,L}), \forall n \in [N]$, by linearity we have that $\overline{W}_{\mathbf{a},[H]} := (\overline{W}_{\mathbf{a},1}, \ldots, \overline{W}_{\mathbf{a},H})$ is the MDS coded version of $W_{\mathbf{a},[L]} := (W_{\mathbf{a},1}, \ldots, W_{\mathbf{a},L}), \forall\, \mathbf{a} \in \mathbb{F}_q^N$, as in (4b).

The system operates in two phases as follows.

Placement Phase: The servers can communicate with each other, and all users can access all servers. To ensure the security condition in (10b), 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 Z_k as a function of P_k, V and the file library $W_{[N]}$. Let the cached content be

$$Z_k := \varphi_k(P_k, V, W_{[N]}),\ \forall k \in [K], \tag{5}$$

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

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

$$H(\mathbf{d}_{[K]}, W_{[N]}, P_{[K]}, V) = \sum_{k \in [K]} H(\mathbf{d}_k) + \sum_{n \in [N]} H(W_n) + \sum_{k \in [K]} H(P_k) + H(V). \tag{6}$$

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

$$\text{as } Q_{k,h} := \kappa_{k,h}(\mathbf{d}_k, Z_k) \in \mathbb{F}_q^{\ell_{k,h}}, \forall h \in [H], \tag{7}$$

for some query functions $\kappa_{k,h} : \mathbb{F}_q^N \times \mathbb{F}_q^{\lfloor MB \rfloor} \mapsto \mathbb{F}_q^{\ell_{k,h}}$, 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

$$X_h := \phi_h(V, Q_{[K],h}, \overline{W}_{[N],h}), \tag{8}$$

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 load of the system is defined as

$$R := \sum_{h \in [H]} R_h. \tag{9}$$

A Robust Secure and (user- and server-) Private cache-aided Linear Function Retrieval (RSP-LFR) scheme must satisfy

[Correctness] $\quad H(W_{\mathbf{d}_k} \mid X_{\mathcal{L}}, \mathbf{d}_k, Z_k) = 0, \quad \forall\, k \in [K], \mathcal{L} \subseteq [H] : |\mathcal{L}| = L,$ (10a)

[Security] $\quad I(W_{[N]}; X_{[H]}) = 0,$ (10b)

[Server Privacy] $\quad I(\mathbf{d}_{[K]}; Q_{[K],[H]}, \overline{W}_{[N],[H]}, V) = 0,$ (10c)

[User Privacy] $\quad I(\mathbf{d}_{[K]\setminus\mathcal{S}}; X_{[H]}, \mathbf{d}_{\mathcal{S}}, Z_{\mathcal{S}} \mid W_{[N]}) = 0, \quad \forall\, \mathcal{S} \subseteq [K] : \mathcal{S} \neq \emptyset.$ (10d)

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

$$R^*(M) := \inf_{B \in \mathbb{N}^+} \big\{ R : (M, R) \text{ is achievable for } B \big\}. \tag{11}$$

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).

Remark 1 (Implications of the conditions in (10)). The constrains (10a)–(10d) imply the following:

1) The correctness condition in (10a) guarantees that each

user can correctly decode its required 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 (10b) guarantees that a wiretapper, who is not a user in the system and observes the delivery signals, can not obtain any information about the contents of the library files. It was proved in [7, Appendix A] that the conditions in (10b) and (10d) 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 server-privacy condition in (10c) 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.
4) The user-privacy condition in (10d) 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.

Remark 2 (Minimum memory size). It was proved in [5] that, in order to guarantee the correctness condition in (10a) and the security condition in (10b) 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 (File retrieval). If the demands $\mathbf{d}_1, \ldots, \mathbf{d}_K$ are restricted to $\{\mathbf{e}_1, \ldots, \mathbf{e}_N\}$, 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, then each user is interested in retrieving one single file.

Remark 4 (Comparison with [12]). For case $L = 1$ and $G = [1, 1, \ldots, 1]$, the servers store replicated databases. A scheme to retrieve a single file per user from replicated databases while guaranteeing server-privacy was proposed in [12]. This differs from our setup, even if we remove the user-privacy and security conditions, since we impose robustness, i.e., each user can decode from the signal received from of any one server (i.e., $L = 1$). Our set-up does not reduce to the PIR setting in [?] again because of the robustness constraint.

III. Main Result and an Example

The following theorem is our main result, achieved by the *Key Superposition RSP-LFR* scheme described in Section IV.

Theorem 1. *For an (N, K, L, H) system, the lower convex envelope of the following points is achievable*

$$(M_t, R_t) = \left(1 + \frac{t(N-1)}{K}, \frac{H(K-t)}{L(t+1)}\right), \ t \in [0 : K]. \quad (12)$$

Moreover, this load-memory tradeoff is optimal to within a multiplicative gap of at most 8 in all regimes, except for $M \in [1, 2)$ with $N < K$.

Remark 5 (Comparison with single-server system). If $H = L = 1$, the system degrades to single-server shared-link system, where all the files are stored at the server [1]. In [7], a key superposition scheme was proposed to guarantee the correctness, security, and user privacy conditions simultaneously. The scheme was presented in the Placement Delivery Array (PDA) framework, which was proposed in [13] to find out low subpacketization schemes. In particular, it was showed that schemes based on PDAs describing the MAN scheme achieve the lower convex envelope of the points

$$(M_t, R'_t) = \Big(1 + \frac{t(N-1)}{K}, \frac{K-t}{t+1}\Big), \quad t \in [0 : K]. \quad (13)$$

Notice that, when $H/L = 1$, the memory-load pairs in (12) degrade to (13). In this case, 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-privacy conditions, the server-privacy condition does not increase the load-memory tradeoff. Moreover, let $R(M)$ denote the tradeoff in (12) and $R_{\text{Signle}}(M)$ the tradeoff in (13), then $R(M) = \frac{H}{L} \cdot R_{\text{Single}}(M)$, i.e., the achieved load is multi-server systems is the single-server load scaled by the inverse of the MDS code rate.

We conclude this section with an example to highlight the key ideas in the RSP-LFR scheme described in Section IV.

Example 1. Consider the $(N, K, L, H) = (4, 3, 2, 3)$ system with MDS generator matrix

$$G = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}. \quad (14)$$

That is, each file $W_n, n \in [N], N = 4$, is split into $L = 2$ subfiles as $W_n = (W_{n,1}, W_{n,2})$, and the contents stored at the $H = 3$ servers are $\overline{W}_{[4],1} = W_{[4],1}$, $\overline{W}_{[4],2} = W_{[4],2}$ and $\overline{W}_{[4],3} = W_{[4],1} \oplus W_{[4],2}$, respectively.

Consider $t = 1$. We partition each subfile $W_{n,l}$ into $\binom{K}{t} = 3$ equal-size packets as $W_{n,l} = (W_{n,l,1}, W_{n,l,2}, W_{n,l,3})$, and as a result, also for each coded subfile $\overline{W}_{n,h} = (\overline{W}_{n,h,1}, \overline{W}_{n,h,2}, \overline{W}_{n,h,3})$, where each packet is in $\mathbb{F}_q^{B/6}$. The system operates as follows:

Placement Phase: The servers share $L\binom{K}{t} = 6$ random packets $\{V_{l,\mathcal{J}} : l \in [2], \mathcal{J} \subseteq [3], |\mathcal{J}| = 2\}$, which are generated independently and uniformly over $\mathbb{F}_q^{B/6}$. Each user $k \in [3]$ generates a random vector $\mathbf{p}_k = (p_{k,1}, p_{k,2}, p_{k,3}, p_{k,4})^\top \in \mathbb{F}_q^4$ and caches

$$Z_k = \{\mathbf{p}_k\} \cup \{W_{n,l,k} : n \in [4], l \in [2]\} \quad (15a)$$
$$\cup \{W_{\mathbf{p}_k,l,j} + V_{l,\{j,k\}} : l \in [2], j \in [3], j \neq k\}. \quad (15b)$$

Delivery Phase: Let the users' demands be $W_{\mathbf{d}_1}, W_{\mathbf{d}_2}$ and $W_{\mathbf{d}_3}$, where $\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3 \in \mathbb{F}_q^4$. Each user k sends the query

$$\mathbf{q}_k = \mathbf{p}_k \oplus \mathbf{d}_k, \quad (16)$$

to all the servers. Upon receiving the queries $\mathbf{q}_{[3]}$, each server $h \in [3]$ sends a signal X_h to the users, which is composed of the queries $\mathbf{q}_{[3]}$ and of the $\binom{K}{t+1} = 3$ coded packets

$$\overline{Y}_{h,\{j,k\}} = \overline{V}_{h,\{j,k\}} + \overline{W}_{\mathbf{q}_j,h,k} + \overline{W}_{\mathbf{q}_k,h,j}, \quad (17)$$

where $\{j, k\} \subseteq [3]$ and $(\overline{V}_{1,\mathcal{J}}, \overline{V}_{2,\mathcal{J}}, \overline{V}_{3,\mathcal{J}})$ is the MDS coded version of $(V_{1,\mathcal{J}}, V_{2,\mathcal{J}})$ for any $\mathcal{J} = \{j, k\} \subseteq [3]$.

Performance: To show that each user k can decode $W_{\mathbf{d}_k}$ with signals from any $L=2$ servers, we notice that for each $\mathcal{J}=\{j,k\}\subseteq[3]$, the packets $(\overline{Y}_{1,\mathcal{J}},\overline{Y}_{2,\mathcal{J}},\overline{Y}_{3,\mathcal{J}})$ are the MDS coded version of $(Y_{1,\mathcal{J}},Y_{2,\mathcal{J}})$, where $Y_{l,\{j,k\}}=V_{l,\{j,k\}}+W_{\mathbf{q}_j,l,k}+W_{\mathbf{q}_k,l,j}$. Thus, upon receiving any $L=2$ of the signals X_1,X_2,X_3, each user k can decode all the coded packets $\{Y_{l,\mathcal{J}}:l\in[2],\mathcal{J}\subseteq[3],|\mathcal{J}|=2\}$. For $j\in[3]\backslash\{k\}$ and $l\in[2]$, user k decodes $W_{\mathbf{d}_k,l,j}$ from $Y_{l,\{j,k\}}$ since

$$Y_{l,\{j,k\}}=W_{\mathbf{d}_k,l,j}+(W_{\mathbf{p}_k,l,j}+V_{l,\{j,k\}})+W_{\mathbf{q}_j,l,k},\quad(18)$$

where $W_{\mathbf{p}_k,l,j}+V_{l,\{j,k\}}$ is cached by user k by (15b), and $W_{\mathbf{q}_j,l,k}$ can be computed by user k from the vector $\mathbf{q}_j$ in (16) and the cache content in (15a). The security condition is guaranteed since the transmitted signal is the coded version of $(Y_{1,\mathcal{J}},Y_{2,\mathcal{J}})$ for each $\mathcal{J}\subseteq[3]$ with cardinality 2, and each signal is added a random vector uniformly distributed over $\mathbb{F}_q^{B/6}$. The server- and user-privacy conditions are guaranteed since the query $\mathbf{q}_k=\mathbf{p}_k+\mathbf{d}_k$ in (16) does not contain any information about $\mathbf{d}_k$, since the vectors $\mathbf{p}_{[K]}$ are independently and uniformly distributed over $\mathbb{F}_q^4$.

Note that each packet is of length $\frac{B}{6}$. Each user caches 12 packets and 1 vector in $\mathbb{F}_q^4$, and each of the servers send 3 packets and 3 vectors in $\mathbb{F}_q^4$. Since the length of vectors in $\mathbb{F}_q^4$ do not scale with B, the achieved memory-load point is $(M,R)=(12\times\frac{1}{6},3\times3\times\frac{1}{6})=(2,\frac{3}{2})$.

IV. Key Superposition RSP-LFR Scheme

Here we describe the *Key Superposition RSP-LFR Scheme* in full generality. The scheme is inspired by the key superposition scheme for the single-server shared-link model [7].

For each $t\in[K]$, define

$$\mathbf{\Omega}_t=\{\mathcal{I}\subseteq[K]\,:\,|\mathcal{I}|=t\}.\quad(19)$$

Notice that for $t=K$ in (12), the achievability of $(M_K,R_K)=(N,0)$ is trivial. In the following, we describe the scheme for $t\in\{0,1,\ldots,K-1\}$.

Firstly, each subfile $W_{n,l}$ is partitioned into $\binom{K}{t}$ equal-size packets, denoted by

$$W_{n,l}=\{W_{n,l,\mathcal{I}}:l\in[L],\mathcal{I}\in\mathbf{\Omega}_t\},\ \forall n\in[N],l\in[L].\quad(20)$$

By (2), each coded subfile $\overline{W}_{n,h}$ is composed of $\binom{K}{t}$ equal-size packets, i.e.,

$$\overline{W}_{n,h}=\{\overline{W}_{n,h,\mathcal{I}}\,:\,\mathcal{I}\in\mathbf{\Omega}_t\},\ \forall\,n\in[N],h\in[H],\quad(21)$$

where $\overline{W}_{n,h,\mathcal{I}}=\sum_{l\in[L]}g_{l,h}W_{n,l,\mathcal{I}}\in\mathbb{F}_q^{B/L}$ for all $\mathcal{I}\in\mathbf{\Omega}_t$, as per (4). The system operates as follows.

Placement Phase: The servers share $L\binom{K}{t+1}$ *security keys* denoted by $\{V_{l,\mathcal{J}}\,:\,l\in[L],\mathcal{J}\in\mathbf{\Omega}_{t+1}\}$, which are independently and uniformly distributed over $\mathbb{F}_q^{B/(L\binom{K}{t})}$. Each user $k\in[K]$ generates a vector $\mathbf{p}_k$ randomly and uniformly from $\mathbb{F}_q^N$, and constructs $\binom{K-1}{t}$ *privacy keys* $\{W_{\mathbf{p}_k,l,\mathcal{I}}:l\in[L],\mathcal{I}\in\mathbf{\Omega}_t,k\notin\mathcal{I}\}$. User $k\in[K]$ caches

$$Z_k=\{\mathbf{p}_k\}\cup\{W_{n,l,\mathcal{I}}:n\in[N],l\in[L],\mathcal{I}\in\mathbf{\Omega}_t,k\in\mathcal{I}\}\quad(22a)$$
$$\cup\{W_{\mathbf{p}_k,l,\mathcal{I}}+V_{l,\mathcal{I}\cup\{k\}}:l\in[L],\mathcal{I}\in\mathbf{\Omega}_t,k\notin\mathcal{I}\}.\quad(22b)$$

Delivery Phase: Assume the demand vector of user $k\in[K]$ is $\mathbf{d}_k$, $\forall\,k\in[K]$. User $k\in[K]$ generates a query vector $\mathbf{q}_k=\mathbf{p}_k+\mathbf{d}_k$, and sends it to all the servers, i.e.,

$$Q_{k,1}=\ldots=Q_{k,H}=\mathbf{q}_k=\mathbf{p}_k+\mathbf{d}_k,\quad\forall\,k\in[K].\ (23)$$

For each $\mathcal{J}\in\mathbf{\Omega}_{t+1}$, let $(\overline{V}_{1,\mathcal{J}},\ldots,\overline{V}_{H,\mathcal{J}})$ be the MDS coded version of $(V_{1,\mathcal{J}},\ldots,V_{L,\mathcal{J}})$ with generator matrix G, i.e., $\overline{V}_{h,\mathcal{J}}=\sum_{l\in[L]}g_{l,h}V_{l,\mathcal{J}}$ for all $h\in[H]$. Upon receving the queries $\mathbf{q}_1,\ldots,\mathbf{q}_K$, server $h\in[H]$ creates a signal for each $\mathcal{J}\in\mathbf{\Omega}_{t+1}$, i.e.,

$$\overline{Y}_{h,\mathcal{J}}:=\overline{V}_{h,\mathcal{J}}+\sum_{j\in\mathcal{J}}\overline{W}_{\mathbf{q}_j,h,\mathcal{J}\backslash\{j\}},\quad\forall\,h\in[H].\quad(24)$$

Notice that $(\overline{Y}_{1,\mathcal{J}},\ldots,\overline{Y}_{H,\mathcal{J}})$ is the MDS coded version of the $(Y_{1,\mathcal{J}},\ldots,Y_{L,\mathcal{J}})$ with the generator matrix G, where the signals $Y_{l,\mathcal{J}}$ are defined as

$$Y_{l,\mathcal{J}}\triangleq V_{l,\mathcal{J}}+\sum_{j\in\mathcal{J}}W_{\mathbf{q}_j,l,\mathcal{J}\backslash\{j\}},\quad\forall\,l\in[L].\quad(25)$$

Server $h\in[H]$ sends the signal

$$X_h=\{\mathbf{q}_k:k\in[K]\}\cup\{\overline{Y}_{h,\mathcal{J}}:\mathcal{J}\in\mathbf{\Omega}_{t+1}\}\quad(26)$$

to the users via its dedicated shared-link.

Correctness in (10a)*:* 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 $W_{\mathbf{d}_k}$, i.e., all the packets $\{W_{\mathbf{d}_k,l,\mathcal{I}}:l\in[L],\mathcal{I}\in\mathbf{\Omega}_t\}$. In fact, for each $\mathcal{I}\in\mathbf{\Omega}_t$ such that $k\in\mathcal{I}$, by (22a), user k has stored all the packets $W_{[N],[L],\mathcal{I}}$, thus it can directly compute the packets $W_{\mathbf{d}_k,l,\mathcal{I}}$ for each $l\in[L]$. Now, consider any $\mathcal{I}\in\mathbf{\Omega}_t$ such that $k\notin\mathcal{I}$. Let $\mathcal{J}=\mathcal{I}\cup\{k\}$, recall that $(\overline{Y}_{1,\mathcal{J}},\ldots,\overline{Y}_{H,\mathcal{J}})$ is the MDS coded version of $(Y_{1,\mathcal{J}},\ldots,Y_{L,\mathcal{J}})$ with generator matrix G, by the property of MDS code, each user can decode all the L coded packets in (25) with any L of the signals $\overline{Y}_{1,\mathcal{J}},\ldots,\overline{Y}_{H,\mathcal{J}}$. Notice that since $\mathcal{J}=\mathcal{I}\cup\{k\}$, the signal $Y_{l,\mathcal{J}}$ is given by

$$Y_{l,\mathcal{I}\cup\{k\}}=V_{l,\mathcal{I}\cup\{k\}}+W_{\mathbf{q}_k,l,\mathcal{I}}+\sum_{j\in\mathcal{I}}W_{\mathbf{q}_j,l,\mathcal{I}\cup\{k\}\backslash\{j\}}\quad(27a)$$
$$=W_{\mathbf{d}_k,l,\mathcal{I}}+V_{l,\mathcal{I}\cup\{k\}}+W_{\mathbf{p}_k,l,\mathcal{I}}+\sum_{j\in\mathcal{I}}W_{\mathbf{q}_j,l,\mathcal{I}\cup\{k\}\backslash\{j\}},\ (27b)$$

where (27b) follows from $\mathbf{q}_k=\mathbf{p}_k+\mathbf{d}_k$. Therefore, user k can decode $W_{\mathbf{d}_k,l,\mathcal{I}}$ from the the signal $Y_{l,\mathcal{I}\cup\{k\}}$ by canceling the remaining terms since

1) the coded packet $V_{l,\mathcal{I}\cup\{k\}}+W_{\mathbf{p}_k,l,\mathcal{I}}$ is cached by user k by (22b);
2) for each $j\in\mathcal{I}$, since $k\in\mathcal{I}\cup\{k\}\backslash\{j\}$, user k can compute $W_{\mathbf{q}_j,l,\mathcal{I}\cup\{k\}\backslash\{j\}}$ from the vector $\mathbf{q}_j$ and the cached packets $W_{[N],l,\mathcal{I}\cup\{k\}\backslash\{j\}}$ by (22a).

Remark 6 (Robustness of Decoding). From the above decoding process, user k can decode its demanded linear function if for any $\mathcal{I}\in\mathbf{\Omega}_t$ such that $k\notin\mathcal{I}$, user k can receive any L of the coded signals $\overline{Y}_{1,\mathcal{I}\cup\{k\}},\ldots,\overline{Y}_{H,\mathcal{I}\cup\{k\}}$. 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 $\mathcal{L}$ of signals varying over different transmissions; and (ii) each user $k \in [K]$ only needs to decode packets over the signals assocatated to $\mathcal{J} \in \mathbf{\Omega}_{t+1}$ such that $k \in \mathcal{J}$.

Security in (10b)*:* We have

$$
\begin{aligned}
& I(W_{[N]}; X_{[H]}) && \text{(28a)}\\
&= I(W_{[N]}; \mathbf{q}_{[K]}, \{\overline{Y}_{h,\mathcal{J}}\}_{h\in[H],\mathcal{J}\in\mathbf{\Omega}_{t+1}}) && \text{(28b)}\\
&= I(W_{[N]}; \mathbf{q}_{[K]}, \{Y_{l,\mathcal{J}}\}_{l\in[L],\mathcal{J}\in\mathbf{\Omega}_{t+1}}) && \text{(28c)}\\
&= I(W_{[N]}; \mathbf{q}_{[K]}) + I(W_{[N]}; \{Y_{l,\mathcal{J}}\}_{l\in[L],\mathcal{J}\in\mathbf{\Omega}_{t+1}} \,|\, \mathbf{q}_{[K]}) && \text{(28d)}\\
&= 0, && \text{(28e)}
\end{aligned}
$$

where: (28c) holds since $(\overline{Y}_{1,\mathcal{J}}, \dots, \overline{Y}_{H,\mathcal{J}})$ is the MDS coded version of $(Y_{1,\mathcal{J}}, \dots, Y_{L,\mathcal{J}})$ for each $\mathcal{J} \in \mathbf{\Omega}_{t+1}$, and hence they determine each other; and (28e) follows since (a) the vectors $\mathbf{q}_{[K]} = \mathbf{d}_{[K]} + \mathbf{p}_{[K]}$ are independent of $W_{[N]}$, and (b) $\{Y_{l,\mathcal{J}}\}_{l\in[L],\mathcal{J}\in\mathbf{\Omega}_{t+1}}$ are independent of $(W_{[N]}, \mathbf{q}_{[K]})$ because the random variables $\{V_{l,\mathcal{J}}\}_{l\in[L],\mathcal{J}\in\mathbf{\Omega}_{t+1}}$ are independently and uniformly distributed over $\mathbb{F}_q^{B/(L\binom{K}{t})}$.

Server Privacy in (10c)*:* We have

$$
\begin{aligned}
& I(\mathbf{d}_{[K]}; Q_{[K],[H]}\, \overline{W}_{[N],[H]}, V) && \text{(29a)}\\
&= I(\mathbf{d}_{[K]}; \mathbf{q}_{[K]}, W_{[N]}, V) && \text{(29b)}\\
&= I(\mathbf{d}_{[K]}; W_{[N]}, V) + I(\mathbf{d}_{[K]}; \mathbf{q}_{[K]} \,|\, W_{[N]}, V) && \text{(29c)}\\
&= 0, && \text{(29d)}
\end{aligned}
$$

where: (29b) follows from (23) and the fact $\overline{W}_{[N],[H]}$ and $W_{[N]}$ determines each other; and (29d) follows from (6) and the fact $\mathbf{q}_{[K]} = \mathbf{p}_{[K]} + \mathbf{d}_{[K]}$ are independent of $(\mathbf{d}_{[K]}, W_{[N]}, V)$ since the vectors $\mathbf{p}_{[K]}$ are independent random variables uniformly distributed over $\mathbb{F}_q^N$.

User Privacy in (10d)*:* We have

$$
\begin{aligned}
& I(\mathbf{d}_{[K]\setminus\mathcal{S}}; Z_{\mathcal{S}}, X_{[H]}, \mathbf{d}_{\mathcal{S}} \,|\, W_{[N]}) && \text{(30a)}\\
&= I(\mathbf{d}_{[K]\setminus\mathcal{S}}; Z_{\mathcal{S}}, \mathbf{q}_{[K]}, \{Y_{l,\mathcal{J}}\}_{l\in[L],\mathcal{J}\in\mathbf{\Omega}_{t+1}}, \mathbf{d}_{\mathcal{S}} \,|\, W_{[N]}) && \text{(30b)}\\
&= 0, && \text{(30c)}
\end{aligned}
$$

where: (30b) follows along similar lines as (28a)–(28c); and (30c) follows since $\mathbf{d}_{[K]\setminus\mathcal{S}} = \mathbf{q}_{[K]\setminus\mathcal{S}} - \mathbf{p}_{[K]\setminus\mathcal{S}}$ is independent of $(Z_{\mathcal{S}}, W_{[N]}, \mathbf{q}_{[K]}, \mathbf{d}_{\mathcal{S}}, \{\overline{Y}_{h,\mathcal{J}}\}_{h\in[H],\mathcal{J}\in\mathbf{\Omega}_{t+1}})$ since $\mathbf{p}_{[K]\setminus\mathcal{S}}$ are independently and uniformly distributed over $\mathbb{F}_q^N$.

Performance: By (20), each subfile is split into $\binom{K}{t}$ equal-size packets, each of length $B/(L\binom{K}{t})$. By the cached content in (22), each user k caches $NL\binom{K-1}{t-1}$ uncoded packets in (22a), $L\binom{K-1}{t}$ coded packets in (22b) and a vecotor of length N in (22a). Therefore, the achieved memory size is

$$
\begin{aligned}
M &= \inf_{B\in\mathbb{N}^+} \frac{1}{B}\left(\frac{\left(NL\binom{K-1}{t-1} + L\binom{K-1}{t}\right)B}{L\binom{K}{t}} + N\right) && \text{(31)}\\
&= 1 + \frac{t(N-1)}{K}. && \text{(32)}
\end{aligned}
$$

Moreover, by (26), each server sends K vectors of length N and $\binom{K}{t+1}$ coded packets, thus the load is given by

$$
R = \inf_{B\in\mathbb{N}^+} \frac{H}{B}\left(\frac{\binom{K}{t+1}B}{L\binom{K}{t}} + KN\right) = \frac{H(K-t)}{L(t+1)}. \quad (33)
$$

Note that, although there are some redundant signals over the servers in (26), i.e., the vectors $\mathbf{q}_{[K]}$ are transmitted by all the servers, by the calculation in (33), further reducing the redundancy does not decrease the load, since the load needed to transmit the vectors $\mathbf{q}_{[K]}$ does not scale with B.

Optimality: Let $R^*_{\text{Single}}(M)$ be the optimal load-memory tradeoff for a single-server network with N files and K users, where the correctness, security and user privacy conditions are imposed as in [7]. For any feasible design of caches $Z_{[K]}$ and signals $X_{[H]}$ in our setup, for any $\mathcal{L} \subseteq [H]$, the contents $Z_{[K]}$ and signal $X \triangleq X_{\mathcal{L}}$ are a feasible scheme in the single server setup. Thus, $\frac{H(X_{\mathcal{L}})}{B} \geq R^*_{\text{Single}}(M)$ holds for all $\mathcal{L} \subseteq [K]$, $|\mathcal{L}| = L$. Therefore,

$$
\begin{aligned}
R^*(M) &\geq \frac{1}{B}\sum_{h\in[H]} H(X_h) = \frac{H}{B}\cdot\frac{1}{H}\sum_{h\in[H]} H(X_h) && \text{(34a)}\\
&\geq \frac{H}{B}\cdot\frac{1}{\binom{H}{L}}\sum_{\mathcal{L}\subseteq[H],|\mathcal{L}|=L} \frac{H(X_{\mathcal{L}})}{L} && \text{(34b)}\\
&= H\cdot\frac{1}{\binom{H}{L}}\sum_{\mathcal{L}\subseteq[H],|\mathcal{L}|=L} \frac{R^*_{\text{Single}}(M)}{L} && \text{(34c)}\\
&\geq \frac{H}{L}\cdot R^*_{\text{Single}}(M), && \text{(34d)}
\end{aligned}
$$

where (34b) follows from Han's inequality [14]. Recall that $R(M) = \frac{H}{L}R_{\text{Single}}(M)$ (see Remark 5), hence by (34d),

$$
\frac{R(M)}{R^*(M)} \leq \frac{R_{\text{Single}}(M)}{R^*_{\text{Single}}(M)}. \quad (35)
$$

Thus, the claimed multiplicative gap result directly follows from (35) and the bound for $\frac{R_{\text{Single}}(M)}{R^*_{\text{Single}}(M)}$ in [7, Theorem 3].

Remark 7 (Special Cases)*.* Note that

1) If the security keys are removed (i.e., setting $V_{\mathcal{J}} = \mathbf{0}$ for all $\mathcal{J} \in \mathbf{\Omega}_{t+1}$), then the scheme degrades to an LFR scheme only guaranteeing server- and user- privacy, which achieves the same memory-load pair as in (12);
2) If the privacy keys are removed (i.e., setting $\mathbf{p}_1 = \ldots = \mathbf{p}_K = \mathbf{0}$), then the scheme degrades to a LFR scheme that only guarantees security, which achieves the same memory-load pair as in (12);
3) If both the security and privacy keys are removed (i.e., setting $V_{\mathcal{J}} = \mathbf{0}$ for all $\mathcal{J} \in \mathbf{\Omega}_{t+1}$ and $\mathbf{p}_1 = \ldots = \mathbf{p}_K = \mathbf{0}$), then the scheme degrades to an ordinary LFR scheme, which achieves the memory-load pair $\left(\frac{tN}{K}, \frac{H(K-t)}{L(t+1)}\right)$.

Remark 8 (Improvement in Special Cases)*.* In the special cases that the security keys (i.e., cases 1) and 3) in Remark 7) are not used, in the regime $N \leq K$, and $t \leq K - N$, some redundant signals determined by the queries $\mathbf{q}_1, \dots, \mathbf{q}_K$ can be removed from each server, similar to the single server cases [4], [6], and better memory-load tradeoff can be achieved.

Acknowledgment

This work was supported in part by NSF Award 1910309.

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, “An index coding approach to caching with uncoded cache placement,” *IEEE Trans. Inf. Theory,* vol. 66, no. 3, pp. 1318–1332, Mar. 2020.

[3] 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, pp. 1281–1296, Feb. 2018.

[4] K. Wan, H. Sun, M. Ji, D. Tuninetti, and G. Gaire, “On the optimal load-memory tradoeff of cache-aided scaler linear function retrieval,” arXiv:2001.03577v1.

[5] 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.

[6] Q. Yan, and D. Tuninetti, “Fundamental limits of caching for demand privacy against colluding users,” arXiv:2008.03642.

[7] Q. Yan, and D. Tuninetti, “Key superposition simultaneously achieves security and privacy in cache-aided linear function retrieval,” arXiv:2009.06000.

[8] 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.

[9] 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.

[10] 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 Secur.* vol. 15, pp. 1248-1260, 2020.

[11] 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.

[12] X. Zhang, K. Wan, H. Sun and M. Ji, “Cache-aided multiuser private information retrieval,” *In proc. 2019 IEEE Int. Sym. Inf. Theory (ISIT)*, Los Angeles, CA, U.S.A, Jun. 2020.

[13] 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.

[14] T. M. Cover and J. A. Thomas, “Elements of Information Theory,” *John Wiley & Sons,* 2012.