Symmetric Private Information Retrieval from MDS Coded Distributed Storage with Non-colluding and Colluding Servers

Qiwen Wang, Mikael Skoglund

Abstract—A user wants to retrieve a file from a database without revealing the identity of the file retrieved to the operator of the database (server), which is known as the problem of *private information retrieval* (PIR). If it is further required that the user obtains no information about the other files in the database, the concept of *symmetric private information retrieval* (SPIR) is introduced to guarantee privacy for both parties. For SPIR, the server(s) need to access some randomness independent of the database, to protect the content of undesired files from the user. The information-theoretic capacity of SPIR is defined as the maximum number of information bits of the desired file retrieved per downloaded bit. In this paper, the problem of SPIR is studied for a distributed storage system with N servers (nodes), where all data (including the files and the randomness) are stored in a distributed way. Specifically, the files are stored by an (N, K_C) -MDS storage code. The randomness is distributedly stored such that any K_C servers store independent randomness information. We consider two scenarios regarding to the ability of the storage nodes to cooperate. In the first scenario considered, the storage nodes do not communicate or collude. It is shown that the SPIR capacity for MDS-coded storage (hence called MDS-SPIR) is $1 - \frac{K_C}{N}$, when the amount of the total randomness of distributed nodes (unavailable at the user) is at least $\frac{K_C}{N-K_C}$ times the file size. Otherwise, the MDS-SPIR capacity equals zero. The second scenario considered is the T -colluding SPIR problem (hence called TSPIR). Specifically, any T out of N servers may collude, that is, they may communicate their interactions with the user to guess the identity of the requested file. In the special case with $K_C = 1$, *i.e.*, the database is replicated at each node, the capacity of TSPIR is shown to be $1 - \frac{T}{N}$, with the ratio of the total randomness size relative to the file size be at least $\frac{T}{N-T}$. For TSPIR with MDS-coded storage (called MDS-TSPIR for short), when restricted to schemes with additive randomness where the servers add the randomness to the answers regardless of the queries received, the capacity is proved to equal $1 - \frac{K_C+T-1}{N}$, with total randomness at least $\frac{K_C+T-1}{N-K_C-T+1}$ times the file size. The MDS-TSPIR capacity for general schemes remains an open problem.

I. INTRODUCTION

Considering the scenario that a user wants to retrieve a file from a database stored remotely at a server, the nature of the data might be privacy-sensitive, for example medical records, stock prices *etc.*, such that the user does not want to reveal the identity of the file retrieved to the server operator. In order to protect the identity of requested files, private information retrieval (PIR) is studied at first in [1], [2] to guarantee user privacy. It is shown that if the database is stored at a single server, the only possible scheme to guarantee information-theoretic user privacy is to download the entire database [1], [2], which is inefficient in practice. It is further shown that the communication cost can be reduced in sublinear scale by replicating the database at multiple non-colluding servers [2]. To further protect the privacy of the database, symmetric private information retrieval (SPIR) is introduced [3], such that in the process of data retrieval the user obtains no more information regarding the database other than the requested file. An example of the application is medical data retrieval, it can be reasonable to require that the user does not learn about other users' medical records from the downloads. Inspired by [1]–[3], the problem of PIR has been widely studied in the theoretical computer science literature, surveyed in [4]. In those works, the problem is studied by considering a file as a single bit and the database as a bit string. The retrieval process includes a querying phase where the user sends queries to the nodes, and a downloading phase where the nodes generate answers after receiving the queries and send back to the user. The objective is to minimize the total communication cost during both the querying phase and the downloading phase.

When the file size is significantly large and the target is to minimize the communication cost of only the downloading phase, the metric of the downloading cost is defined as the number of bits downloaded per bit of the retrieved file, and the reciprocal of which is named the *retrieval rate*. The supremum of all achievable retrieval rates is defined as the *PIR capacity*. A series of recent works derives information-theoretic limits of various variants of the PIR problem [5]–[9] *etc.* The leading work in the area is by Sun and Jafar [5], where the authors characterize the capacity of the PIR problem with replicated databases. In subsequent works by Sun and Jafar [6], [7], the PIR capacity with replicated databases, replicated databases with colluding servers, SPIR with replicated (non-colluding) databases are derived. In [8], [9], Banawan and Ulukus find the capacity of the PIR problem with MDS-coded databases,

Manuscript received January 5, 2018; revised December 19, 2018; accepted February 13, 2019. This paper was presented in part at the 2017 IEEE International Conference on Communications and the 2017 IEEE Information Theory Workshop..

This work was supported by the Knut and Alice Wallenberg Foundation and the Swedish Research Council.

Q. Wang is with the Department of Information Science and Engineering, KTH Royal Institute of Technology, SE-100 44, Stockholm, Sweden. (email: qiwenw@kth.se).

M. Skoglund is with the Department of Information Science and Engineering, KTH Royal Institute of Technology, SE-100 44, Stockholm, Sweden. (email: skoglund@ee.kth.se).

Copyright (c) 2017 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

and multi-message PIR with replicated databases.

Considering the aspect of cost in storage systems, replicating the database results in low repair cost for node damage, but with the expense of high storage cost. Coded storage is proposed to utilize the tradeoff between storage cost and repair cost [10]. By using erasure codes, each node only stores a fraction of the whole database, hence reducing the storage cost. The first work on PIR for coded databases known to us appears in [11]. In [11], the authors show that by downloading one extra bit besides the amount of the file size, user privacy can be guaranteed. However, to achieve this low downloading cost, the number of storage nodes needs to grow with the file size, which can be impractical in some storage systems. Later, [12] also considers PIR with coded storage, and focuses on reducing the storage overhead by increasing the number of storage nodes. In [13], PIR for coded databases is investigated, and the tradeoff between storage cost and downloading cost is analyzed. Subsequently in [14], explicit storage and communication schemes to achieve PIR with MDS storage codes are presented, matching the tradeoff derived in [13]. It is worth noting that in the recent work of [8], the capacity of PIR for coded databases is settled, which improves the results in [13] and [14]. Recently in [15], the authors present a framework for PIR from coded databases with colluding servers, which is the PIR version of the problem considered in this work. In [15], a conjecture on the capacity of the problem is proposed, which is settled more recently in [16] through some counterexamples. The PIR capacity with coded databases and colluding servers remains open.

In this work, the problem of SPIR is studied for distributed storage systems with MDS storage codes, where all data are MDS-coded and distributed among the servers. Two scenarios are considered with respect to the ability of the nodes to collude: coded SPIR with non-colluding servers (MDS-SPIR), and coded SPIR with colluding servers (MDS-TSPIR). We show that in order to guarantee SPIR in the non-trivial context, *e.g.*, the number of files K in the database is greater than or equal to 2, the storage nodes need to access some randomness which is independent from the database and meanwhile unavailable to the user. This result is in analogy with that in [7] for the uncoded database. We consider both the files and the randomness as data which are distributed in the MDS storage system at the same phase.¹ The database (files) is stored by an (N, K_C) -MDS code; the randomness is also stored in a distributed way, where for any K_C servers, the randomness stored at the servers are independent. The capacity of MDS-SPIR is found to be $1 - \frac{K_C}{N}$, with the amount of total randomness at least $\frac{K_C}{N-K_C}$ times the file size. Replicated databases is a special case of coded databases with $(N, 1)$ -MDS code. Therefore, our result includes that in [7] for replicated databases as a special case with $K_C = 1$. For the second scenario with T -colluding servers, any T out of N servers may collude, that is, they may communicate their interactions with the user to guess the identity of the requested file. We first consider the special case with $K_C = 1$ for replicated databases

(hence called TSPIR). The capacity of TSPIR is derived to be $1 - \frac{T}{N}$, with the ratio of total randomness size relative to the file size be at least $\frac{T}{N-T}$. This is a generalization of the SPIR capacity in [7] for colluding servers, and includes the result in [7] as a special case with $T = 1$. For TSPIR with coded database (named MDS-TSPIR), we analyze the set of schemes with additive randomness, where the servers add the randomness to the answers regardless of what queries are received. We show that for the additive randomness schemes, the capacity of MDS-TSPIR equals $1 - \frac{K_C+T-1}{N}$, if the total amount of randomness is at least $\frac{K_C+T-1}{N-K_C-T+1}$ times the file size. MDS-TSPIR capacity for general schemes remains an open problem. Table I summarizes the results from previous works and derived in this work on the PIR and SPIR problems with coded-storage and/or colluding databases.

II. MODEL

A. Notations

Let $[m : n]$ denote the set $\{m, m+1, \dots, n\}$ for $m \leq n$. For the sake of brevity, denote the ordered set of random variables $\{X_m, X_{m+1}, \dots, X_n\}$ by $X_{[m:n]}$. The set of random variables $\{X_i : i \in \mathcal{S}\}$ with index set $\mathcal{S}$ is denoted by $X_{\mathcal{S}}$. The transpose of matrix $\mathbf{A}$ is denoted by $\mathbf{A}^T$.

B. Problem Description

Database: A database comprises K independent files, denoted by $W_1, \dots, W_K$. Each file consists of L symbols drawn independently and uniformly from the finite field $\mathbb{F}_q$. Therefore, for any $k \in [1 : K]$,

$$H(W_k) = L \log q \quad ; \quad H(W_1, \dots, W_K) = KL \log q.$$

Database storage: The database is stored in a distributed storage system consisting of N servers (nodes) by an (N, K_C) -MDS storage code. The data stored at node- n is denoted by D_n . Note that for any K_C nodes $\{n_1, \dots, n_{K_C}\} \subset [1 : N]$, the data $\{D_{n_1}, \dots, D_{n_{K_C}}\}$ are linearly and stochastically independent. Furthermore, they can exactly recover the whole database, *i.e.*,

$$H(D_{n_1}, \dots, D_{n_{K_C}}) = H(W_1, \dots, W_K) = KL \log q, \\ H(W_1, \dots, W_K | D_{n_1}, \dots, D_{n_{K_C}}) = 0.$$

In the special case where $K_C = 1$, each node stores the whole set of all K files, which we refer to as *replicative storage*.

Randomness: Each node n stores a random variable S_n , which is independent of the files and unavailable to the user. The random variable S_n represents the information needed in generating the answer from the scheme followed by node n and utilized to protect database-privacy (6) below. Denote all the randomness by $S = \{S_1, S_2, \dots, S_N\}$, where for any K_C nodes $\mathcal{M} \subset [1 : N]$, the K_C random variables $\{S_n : n \in \mathcal{M}\}$ should be statistically independent. This property is required in correspondence to the distributed storage of the database, and can be implemented by choosing K_S statistically independent (but not necessarily uniformly distributed) variables $S_1, \dots, S_{K_S}$ (where $K_S \geq K_C$ is a parameter that can

¹In Section VIII, we discuss the case where the complete randomness information is available to all servers before the distribution of the files, which results in a fundamentally different problem.

	PIR	SPIR
Replicative storage non-colluding	$C_{\text{PIR}} = \frac{1 - \frac{1}{N}}{1 - (\frac{1}{N})^K}$ [5]	$C_{\text{SPIR}} = \begin{cases} 1 - \frac{1}{N}, & \text{if } \rho \geq \frac{1}{N-1} \\ 0, & \text{otherwise} \end{cases}$ [7]
Coded storage non-colluding	$C_{\text{MDS-PIR}} = \frac{1 - \frac{K_C}{N}}{1 - (\frac{K_C}{N})^K}$ [8]	$C_{\text{MDS-SPIR}} = \begin{cases} 1 - \frac{K_C}{N}, & \text{if } \rho \geq \frac{K_C}{N-K_C} \\ 0, & \text{otherwise} \end{cases}$
Replicative storage colluding	$C_{\text{TPIR}} = \frac{1 - \frac{T}{N}}{1 - (\frac{T}{N})^K}$ [6]	$C_{\text{TSPIR}} = \begin{cases} 1 - \frac{T}{N}, & \text{if } \rho \geq \frac{T}{N-T} \\ 0, & \text{otherwise} \end{cases}$
Coded storage colluding	Capacity remains open; partial results in [15], [16]	The capacity in general is an open problem; however, for additive randomness, we have $C_{\text{MDS-TSPIR}}^{+S} = \begin{cases} 1 - \frac{K_C+T-1}{N}, & \text{if } \rho \geq \frac{K_C+T-1}{N-K_C-T+1} \\ & \text{and } N > K_C + T \\ 0, & \text{otherwise} \end{cases}$

TABLE I
A SUMMARY OF RESULTS ON PIR AND SPIR PROBLEMS WITH CODED STORAGE AND/OR COLLUDING DATABASES.

be chosen to optimize the scheme performance) from the same finite field, and then generating $S_1, \dots, S_N$ by using a systematic MDS code.

- For MDS-SPIR, we find that it suffices to set $K_S = K_C$ and use the same (N, K_C) -MDS code for storing both the files and the randomness. In practical systems, it is natural to have this setting when the randomness is a source of data to protect the privacy of the files, and is to be distributed together with the files in the storage system.²
- For TSPIR ($K_C = 1$), we found that it suffices to set $K_S = T$ for T -colluding servers (5).
- For MDS-TSPIR schemes with additive randomness (2), we require that $S_1, \dots, S_N$ are generated from K_S uniformly distributed i.i.d. symbols/vectors by an (N, K_S) -MDS code. We found that when restricted to additive randomness, the value of K_S is not important (referring to Lemma 8).

The total amount of randomness is $H(S) = H(S_1, \dots, S_N)$.

User queries: A user wants to retrieve a file W_κ from the database, where the desired file index κ is drawn uniformly at random from $[1 : K]$. The MDS storage codes are known to the user. In addition to this, the user has no knowledge of the stored data. Let $\mathcal{U}$ denote a random variable privately generated by the user, which represents the randomness of the query scheme followed by the user. The random variable $\mathcal{U}$ is independent of the database, the randomness and the desired file index. Based on the realization of the desired file index k and the realization of $\mathcal{U}$, the user generates and sends queries to all nodes, where the query received by node- n is denoted by $Q_n^{[k]}$. Let $\mathcal{Q} = [Q_n^{[k]}]_{n \in [1:N], k \in [1:K]}$ denote the complete query scheme, namely, the collection of all queries under all cases of desired file index. We have that $H(\mathcal{Q}|\mathcal{U}) = 0$.

Node answers: Based on the received query $Q_n^{[k]}$, the stored data D_n , and the randomness S_n , each node sends an answer

$A_n^{[k]}$ to the user. Therefore,

$$H(A_n^{[k]}|Q_n^{[k]}, D_n, S_n) = 0. \quad (1)$$

For MDS-TSPIR, we further require that the randomness S_n available at node n is added to the answers in a fixed way (regardless of the queries received), that is,

$$[\text{Additive randomness}] \quad A_n^{[k]} = F_n^{[k]}(Q_n^{[k]}, D_n) + S_n, \quad (2)$$

where $F_n^{[k]}(Q_n^{[k]}, D_n)$ denotes the output of the function implemented by node n with the query $Q_n^{[k]}$ and stored data D_n as input, and is abbreviated as $F_n^{[k]}$.³

SPIR: Based on the received answers $A_{[1:N]}^{[k]}$ and the query scheme $\mathcal{Q}$, the user shall be able to decode the requested file W_k with zero error. Hence,

$$H(W_k|A_{[1:N]}^{[k]}, \mathcal{Q}) = 0. \quad (3)$$

For the scenario with non-colluding servers, the nodes do not communicate, that is, they share no information regarding other nodes' queries and answers. The constraint below must be satisfied to guarantee user-privacy:

User-privacy with non-colluding servers: the nodes shall not be able to obtain any information regarding the identity of the requested file, i.e.,

$$I(\kappa; Q_n^{[\kappa]}, W_{[1:K]}, S) = 0, \quad \forall n \in [1 : N]. \quad (4)$$

For the case with T -colluding servers, any set of T nodes may collude to guess the requested file index, by communicating their interactions with the user.

User-privacy with T -colluding servers: any T colluding nodes shall not be able to obtain any information regarding the identity of the requested file, i.e.,

$$I(\kappa; Q_{\mathcal{T}}^{[\kappa]}, W_{[1:K]}, S) = 0, \quad \forall \mathcal{T} \subset [1 : N], |\mathcal{T}| = T. \quad (5)$$

²In Section VIII, we discuss another model with the same randomness shared at all servers, which is motivated by applications where the randomness is to be replicated at all servers a priori, or where the servers have access to the same seeds for generating all the randomness.

³The function $F_n^{[k]}$, randomness S_n , and the answer $A_n^{[k]}$ can be vectors, with (2) being vector addition.

For the problem of SPIR, the constraint below shall be satisfied to guarantee privacy at the database side.

Database-privacy: the user shall learn no information regarding other files in the database, that is, defining $W_{\bar{k}} = \{W_1, \dots, W_{k-1}, W_{k+1}, \dots, W_K\}$,

$$I(W_{\bar{k}}; A_{[1:N]}^{[k]}, \mathcal{Q}, \kappa) = 0. \quad (6)$$

We use the same definition as in [7] for the rate and capacity of SPIR schemes.

Definition 1 (SPIR Rate and Capacity). *The rate of an SPIR scheme is the number of information bits of the requested file retrieved per downloaded answer bit. By symmetry among all files, for any $k \in [1 : K]$,*

$$R \triangleq \frac{H(W_k)}{\sum_{n=1}^N H(A_n^{[k]})}.$$

The capacity is the supremum of the rates achieved over all SPIR schemes.

Definition 2 (Secrecy Rate). *The secrecy rate is the total amount of randomness relative to the file size, that is*

$$\rho \triangleq \frac{H(S)}{H(W_1)}.$$

An SPIR scheme is called an *achievable* scheme if the user can successfully decode the desired file, with both user-privacy and database-privacy constraints guaranteed. An achievable SPIR scheme is *optimal* if it achieves the capacity, in other words, the user can decode successfully with the minimum downloading cost, moreover, the nodes use the least amount of total randomness needed to generate the answers.

III. MAIN RESULTS

When there is only one file in the database, the two privacy constraints become trivial. When $K \geq 2$, we summarize the main results on the capacity of MDS-SPIR, TSPIR and additive randomness MDS-TSPIR below.

Theorem 1 (MDS-SPIR Capacity). *The MDS-SPIR capacity for distributed storage systems with (N, K_C) -MDS coded files is*

$$C_{\text{MDS-SPIR}} = \begin{cases} 1 - \frac{K_C}{N}, & \text{if } \rho \geq \frac{K_C}{N-K_C} \\ 0, & \text{otherwise} \end{cases}.$$

Theorem 2 (TSPIR Capacity). *The TSPIR capacity for replicative storage systems with T -colluding servers is*

$$C_{\text{TSPIR}} = \begin{cases} 1 - \frac{T}{N}, & \text{if } \rho \geq \frac{T}{N-T} \\ 0, & \text{otherwise} \end{cases}.$$

Theorem 3 (Additive randomness MDS-TSPIR Capacity). *For MDS-TSPIR with distributive storage systems where files are stored by an (N, K_C) -MDS storage code, and with T -colluding servers, when restricted to schemes where the servers add randomness to the answers that is independent*

of the queries, the capacity is

$$C_{\text{MDS-TSPIR}}^{+S} = \begin{cases} 1 - \frac{K_C+T-1}{N}, & \text{if } \rho \geq \frac{K_C+T-1}{N-K_C-T+1} \\ & \text{and } N > K_C + T \\ 0, & \text{otherwise} \end{cases}.$$

Remark:

- 1) When $K_C = 1$ in Theorem 1 or $T = 1$ in Theorem 2, the result reduces to Theorem 1 in [7] for SPIR with replicated and non-colluding databases.
- 2) In [8], the authors show that the PIR capacity with MDS storage codes is $(1 + \frac{K_C}{N} + \dots + \frac{K_C^{K-1}}{N^{K-1}})^{-1}$. (We translate their result into our notation.) As the number of files K tends to infinity, the MDS-PIR capacity in [8] approaches the MDS-SPIR capacity in Theorem 1.
- 3) It is proved in [6] that the capacity of TPIR is $(1 + \frac{T}{N} + \dots + \frac{T^{K-1}}{N^{K-1}})^{-1}$. Similar as the relation between the previous PIR and SPIR capacities, when $K \rightarrow \infty$, the TPIR capacity approaches the one of TSPIR in Theorem 2.
- 4) For MDS-SPIR, when the capacity is achieved, the randomness can be stored by the same MDS code for the storage of files.
- 5) The problem of MDS-TPIR with no requirement on the database-privacy is directly related to the SPIR version studied in this work. In Theorem 5 in [16], it is shown that if $N < K_C + T$, as $K \rightarrow \infty$ the capacity of MDS-TPIR decays and converges to 0. This matches with our result that to achieve positive rate for MDS-TSPIR, the number of servers N should be at least $K_C + T$ (although within this work we restrict to schemes with additive randomness). The capacity of MDS-TPIR with general parameters and the capacity of MDS-TSPIR for general schemes remain open problems, and are important in understanding the synergistic effect of the two elements of MDS storage and colluding servers.

IV. PRELIMINARIES

Before presenting the detailed proofs of the main theorems, we first state the following lemmas which hold for SPIR with MDS storage regardless of whether the nodes can collude or not, and will be used for the converse proofs in Section V-A and Section VII-A.

Lemma 4. *For any set of nodes $\mathcal{N} \subset [1 : N]$, $\mathcal{Q} \rightarrow Q_{\mathcal{N}}^{[k]} \rightarrow A_{\mathcal{N}}^{[k]}$. Therefore,*

$$H(A_{\mathcal{N}}^{[k]} | \mathcal{Q}, Q_{\mathcal{N}}^{[k]}) = H(A_{\mathcal{N}}^{[k]} | Q_{\mathcal{N}}^{[k]}), \quad (7)$$

$$H(A_{\mathcal{N}}^{[k]} | \mathcal{Q}, W_k, Q_{\mathcal{N}}^{[k]}) = H(A_{\mathcal{N}}^{[k]} | W_k, Q_{\mathcal{N}}^{[k]}). \quad (8)$$

Proof: The proof of (7): For any set of nodes $\mathcal{N} \subset [1 : N]$, $\mathcal{Q} \rightarrow Q_{\mathcal{N}}^{[k]} \rightarrow A_{\mathcal{N}}^{[k]}$ forms a Markov chain. That is, given the queries received, the answers are independent of the other queries (to other nodes and/or with other desired file indices). Therefore, $H(A_{\mathcal{N}}^{[k]} | \mathcal{Q}, Q_{\mathcal{N}}^{[k]}) = H(A_{\mathcal{N}}^{[k]} | Q_{\mathcal{N}}^{[k]})$.

The proof of (8): We first show that $I(A_{\mathcal{N}}^{[k]}; \mathcal{Q} | W_k, Q_{\mathcal{N}}^{[k]}) \leq$

0, as follows

$$\begin{aligned}
& I(A_{\mathcal{N}}^{[k]}; \mathcal{Q} | W_k, Q_{\mathcal{N}}^{[k]}) \\
& \leq I(A_{\mathcal{N}}^{[k]}, W_{[1:K]}, S; \mathcal{Q} | W_k, Q_{\mathcal{N}}^{[k]}) \\
& = I(W_{[1:K]}, S; \mathcal{Q} | W_k, Q_{\mathcal{N}}^{[k]}) + I(A_{\mathcal{N}}^{[k]}; \mathcal{Q} | W_{[1:K]}, S, W_k, Q_{\mathcal{N}}^{[k]}) \\
& \stackrel{(a)}{=} I(W_{[1:K]}, S; \mathcal{Q} | W_k, Q_{\mathcal{N}}^{[k]}) \\
& \leq I(W_{[1:K]}, S; \mathcal{Q}) = 0,
\end{aligned}$$

where (a) holds because the answers are deterministic functions of the database, the randomness, and the queries. In the last step, $I(W_{[1:K]}, S; \mathcal{Q}) = 0$ holds because the queries do not depend on the database and the randomness (1).

On the other hand, it is immediate that $I(A_{\mathcal{N}}^{[k]}; \mathcal{Q} | W_k, Q_{\mathcal{N}}^{[k]}) \geq 0$. Therefore, $H(A_{\mathcal{N}}^{[k]} | W_k, Q_{\mathcal{N}}^{[k]}) = H(A_{\mathcal{N}}^{[k]} | \mathcal{Q}, W_k, Q_{\mathcal{N}}^{[k]})$. $\square$

Similar as in the previous work on SPIR with replicated storage [7], because of user-privacy, the query and answer for each node are symmetric across file indices, and this also holds if conditioned on part of the database. We recapitulate the results in Lemma 5 below.

Lemma 5 (Symmetry across files). *For any node $n \in [1 : N]$, and for any file index $k, k' \in [1 : K]$,*

$$H(A_n^{[k]} | Q_n^{[k]}) = H(A_n^{[k']} | Q_n^{[k']}); \quad (9)$$

$$H(A_n^{[k]} | W_k, Q_n^{[k]}) = H(A_n^{[k']} | W_k, Q_n^{[k']}). \quad (10)$$

V. MDS-SPIR

A. Converse

In this section, we prove the converse part of Theorem 1. That is, to achieve positive SPIR rate with an (N, K_C) -MDS storage code, the ratio of total randomness relative to the file size $\frac{H(S)}{H(W_1)}$ needs to be at least $\frac{K_C}{N-K_C}$, and the user needs to download at least $\frac{N}{N-K_C}L$ symbols. Recall that L is the file size. In the following, we first show Lemmas 6-7, which will be used in the major part of the proof in Section V-A1 and Section V-A2.

Lemma 6. *For any K_C nodes $n_{[1:K_C]} \triangleq \{n_1, \dots, n_{K_C}\} \subset [1 : N]$,*

$$H(A_{n_{[1:K_C]}}^{[k]} | Q_{n_{[1:K_C]}}^{[k]}) = H(A_{n_{[1:K_C]}}^{[k']} | Q_{n_{[1:K_C]}}^{[k']}); \quad (11)$$

$$H(A_{n_{[1:K_C]}}^{[k]} | W_k, Q_{n_{[1:K_C]}}^{[k]}) = H(A_{n_{[1:K_C]}}^{[k']} | W_k, Q_{n_{[1:K_C]}}^{[k']}). \quad (12)$$

Proof: The proof of (11) follows the same arguments as (12). Therefore, in the following we present the proof details of (12). Because the database storage is (N, K_C) -MDS, and every K_C nodes store statistically independent S_n , the information stored at nodes $n_{[1:K_C]}$ is statistically independent (see Lemma 1 in [8] for a formal proof). For any node n , the answer $A_n^{[k]}$ is a deterministic function of the query $Q_n^{[k]}$, the stored database D_n , and the stored randomness S_n . Given the queries and W_k , the randomness of the K_C answers only depends on D_n and S_n for $n \in \{n_1, \dots, n_{K_C}\}$, which are statistically

independent. Therefore,

$$\begin{aligned}
H(A_{n_{[1:K_C]}}^{[k]} | W_k, Q_{n_{[1:K_C]}}^{[k]}) &= \sum_{n=n_1}^{n_{K_C}} H(A_n^{[k]} | W_k, Q_{n_{[1:K_C]}}^{[k]}) \\
&\stackrel{(a)}{=} \sum_{n=n_1}^{n_{K_C}} H(A_n^{[k]} | W_k, Q_n^{[k]}) \\
&\stackrel{(b)}{=} \sum_{n=n_1}^{n_{K_C}} H(A_n^{[k']} | W_k, Q_n^{[k']}),
\end{aligned} \quad (13)$$

where (a) and (b) follow from Lemma 4 and Lemma 5 respectively. Similarly,

$$\begin{aligned}
H(A_{n_{[1:K_C]}}^{[k']} | W_k, Q_{n_{[1:K_C]}}^{[k']}) &= \sum_{n=n_1}^{n_{K_C}} H(A_n^{[k']} | W_k, Q_{n_{[1:K_C]}}^{[k']}) \\
&= \sum_{n=n_1}^{n_{K_C}} H(A_n^{[k']} | W_k, Q_n^{[k']}). \quad (14)
\end{aligned}$$

Combining (13) and (14), we have that $H(A_{n_{[1:K_C]}}^{[k]} | W_k, Q_{n_{[1:K_C]}}^{[k]}) = H(A_{n_{[1:K_C]}}^{[k']} | W_k, Q_{n_{[1:K_C]}}^{[k']})$. $\square$

Lemma 7. *For any optimal scheme, and for any K_C nodes $n_{[1:K_C]} \triangleq \{n_1, \dots, n_{K_C}\} \subset [1 : N]$,*

$$H(A_{n_{[1:K_C]}}^{[k]} | W_k, Q_{n_{[1:K_C]}}^{[k]}) = H(A_{n_{[1:K_C]}}^{[k]} | Q_{n_{[1:K_C]}}^{[k]}).$$

Proof: By database-privacy (6), $I(W_{\bar{k}'}; A_{[1:N]}^{[k']}, \mathcal{Q}) = 0$. For $k \neq k'$, $W_k \in W_{\bar{k}'}$. W.l.o.g., choose the size- K_C node set to be $[1 : K_C]$,

$$\begin{aligned}
0 &= I(W_k; A_{[1:K_C]}^{[k']}, Q_{[1:K_C]}^{[k']}) \\
&= I(W_k; A_{[1:K_C]}^{[k']} | Q_{[1:K_C]}^{[k']}) + I(W_k; Q_{[1:K_C]}^{[k']}) \\
&\stackrel{(a)}{=} I(W_k; A_{[1:K_C]}^{[k']} | Q_{[1:K_C]}^{[k']}) \\
&= H(A_{[1:K_C]}^{[k']} | Q_{[1:K_C]}^{[k']}) - H(A_{[1:K_C]}^{[k']} | W_k, Q_{[1:K_C]}^{[k']}) \\
&\stackrel{(b)}{=} H(A_{[1:K_C]}^{[k']} | Q_{[1:K_C]}^{[k']}) - H(A_{[1:K_C]}^{[k]} | W_k, Q_{[1:K_C]}^{[k]}) \\
&\stackrel{(c)}{=} H(A_{[1:K_C]}^{[k]} | Q_{[1:K_C]}^{[k]}) - H(A_{[1:K_C]}^{[k]} | W_k, Q_{[1:K_C]}^{[k]}),
\end{aligned}$$

where equality (a) holds because W_k is independent of the queries; equalities (b) and (c) follow from Lemma 6. $\square$

1) *The proof for $R \leq C_{\text{MDS-SPIR}}$:* For any file W_k , $k \in [1 : K]$,

$$\begin{aligned}
H(W_k) &= H(W_k | \mathcal{Q}) \\
&\stackrel{(a)}{=} H(W_k | \mathcal{Q}) - H(W_k | A_{[1:N]}^{[k]}, \mathcal{Q}) \\
&= I(W_k; A_{[1:N]}^{[k]} | \mathcal{Q}) \\
&= H(A_{[1:N]}^{[k]} | \mathcal{Q}) - H(A_{[1:N]}^{[k]} | W_k, \mathcal{Q}) \\
&\leq H(A_{[1:N]}^{[k]} | \mathcal{Q}) - H(A_{n_{[1:K_C]}}^{[k]} | W_k, \mathcal{Q}, Q_{n_{[1:K_C]}}^{[k]}) \\
&\stackrel{(b)}{=} H(A_{[1:N]}^{[k]} | \mathcal{Q}) - H(A_{n_{[1:K_C]}}^{[k]} | W_k, Q_{n_{[1:K_C]}}^{[k]}) \\
&\stackrel{(c)}{=} H(A_{[1:N]}^{[k]} | \mathcal{Q}) - H(A_{n_{[1:K_C]}}^{[k]} | Q_{n_{[1:K_C]}}^{[k]})
\end{aligned}$$

$$\leq H(A_{[1:N]}^{[k]}|\mathcal{Q}) - H(A_{n_{[1:K_C]}^{[k]}|\mathcal{Q}),$$

Equality (a) holds follows from (3). Also, (b) and (c) follow from Lemma 4 and Lemma 7.

Averaging over all $n_{[1:K_C]}$ with K_C nodes, we have that

$$H(W_k) \leq H(A_{[1:N]}^{[k]}|\mathcal{Q}) - \frac{1}{\binom{N}{K_C}} \sum_{n_{[1:K_C]} \subset [1:N]} H(A_{n_{[1:K_C]}^{[k]}|\mathcal{Q}).$$

By Han's inequality [17],

$$\frac{1}{\binom{N}{K_C}} \sum_{n_{[1:K_C]} \subset [1:N]} H(A_{n_{[1:K_C]}^{[k]}|\mathcal{Q}) \geq \frac{K_C}{N} H(A_{[1:N]}^{[k]}|\mathcal{Q}).$$

$$\text{Therefore, } R = \frac{H(W_k)}{\sum_{n=1}^N H(A_n^{[k]})} \leq 1 - \frac{K_C}{N}. \quad \square$$

2) *The proof for $\rho \geq \frac{K_C}{N-K_C}$:* By database-privacy (6),

$$\begin{aligned} 0 &= I(W_k; A_{[1:N]}^{[k]}, \mathcal{Q}) \\ &= I(W_k; A_{[1:N]}^{[k]}|\mathcal{Q}) + I(W_k; \mathcal{Q}) \\ &= I(W_k; A_{[1:N]}^{[k]}|\mathcal{Q}) \\ &= H(W_k|\mathcal{Q}) - H(W_k|A_{[1:N]}^{[k]}, \mathcal{Q}) \\ &\stackrel{(a)}{=} H(W_k|\mathcal{Q}, W_k) - H(W_k|A_{[1:N]}^{[k]}, \mathcal{Q}, W_k) \\ &= I(W_k; A_{[1:N]}^{[k]}|\mathcal{Q}, W_k) \\ &\geq I(W_k; A_{n_{[1:K_C]}^{[k]}|\mathcal{Q}, W_k) \\ &\stackrel{(b)}{=} H(A_{n_{[1:K_C]}^{[k]}|\mathcal{Q}, W_k) - H(A_{n_{[1:K_C]}^{[k]}|\mathcal{Q}, W_{[1:K]}) \\ &\quad + H(A_{n_{[1:K_C]}^{[k]}|\mathcal{Q}, W_{[1:K]}, S) \\ &= H(A_{n_{[1:K_C]}^{[k]}|\mathcal{Q}, W_k) - I(S; A_{n_{[1:K_C]}^{[k]}|\mathcal{Q}, W_{[1:K]}) \\ &\geq H(A_{n_{[1:K_C]}^{[k]}|\mathcal{Q}, W_k, Q_{n_{[1:K_C]}^{[k]}}) - H(S) \\ &\stackrel{(c)}{=} H(A_{n_{[1:K_C]}^{[k]}|W_k, Q_{n_{[1:K_C]}^{[k]}}) - H(S) \\ &\stackrel{(d)}{=} H(A_{n_{[1:K_C]}^{[k]}|Q_{n_{[1:K_C]}^{[k]}}) - H(S) \\ &\geq H(A_{n_{[1:K_C]}^{[k]}|\mathcal{Q}) - H(S) \end{aligned}$$

Equality (a) holds because W_k is independent of other files $W_{\bar{k}}$, and from all the answers $A_{[1:N]}^{[k]}$ and the queries $\mathcal{Q}$ the user can decode W_k . Equality (b) holds because the answers $A_{n_{[1:K_C]}^{[k]}}$ are deterministic functions of the queries $\mathcal{Q}$, the database $W_{[1:K]}$, and the randomness S . Equalities (c) and (d) follow from Lemma 4 and Lemma 7.

Averaging over all $n_{[1:K_C]}$, and from the proof in Section V-A1 above,

$$\begin{aligned} H(S) &\geq \frac{1}{\binom{N}{K_C}} \sum_{n_{[1:K_C]} \subset [1:N]} H(A_{n_{[1:K_C]}^{[k]}|\mathcal{Q}) \\ &\geq \frac{K_C}{N} H(A_{[1:N]}^{[k]}|\mathcal{Q}) \\ &\geq \frac{K_C}{N-K_C} H(W_k). \end{aligned}$$

$$\text{Hence, } \rho = \frac{H(S)}{H(W_k)} \geq \frac{K_C}{N-K_C}. \quad \square$$

B. Achievability

In this section, we present a scheme which achieves the maximum SPIR rate and lowest secrecy rate in Section V-A. Specifically, the user is able to decode the desired file successfully and privately by downloading $\frac{N}{N-K_C}L$ symbols, and obtains no further information regarding the database with $\frac{K_C}{N-K_C}L$ uniformly random symbols coded and distributed among the nodes. The achievable scheme is developed from the scheme in [14] by adding randomness to the answers. The main concepts used in the construction are,

- The user hides the identity of the desired file in randomly generated queries, such that the queries appear statistically uniformly random to the nodes.
- The nodes add random symbols that are independent of the database and unavailable to the user in the answers to protect the content of other files. The random symbols are added according to the storage code construction for successful decoding.
- The user downloads the lowest possible number of symbols to construct a linear system that is solvable. The unknowns are symbols of the requested file, and some function outputs generated from queries, stored data and the randomness.

Database: W.l.o.g., assume each file consists of $L = (N - K_C)K_C$ symbols. Specifically,

$$W_k = \begin{bmatrix} w_{k,1}^1 & w_{k,2}^1 & \dots & w_{k,K_C}^1 \\ w_{k,1}^2 & w_{k,2}^2 & \dots & w_{k,K_C}^2 \\ \vdots & \vdots & \ddots & \vdots \\ w_{k,1}^{N-K_C} & w_{k,2}^{N-K_C} & \dots & w_{k,K_C}^{N-K_C} \end{bmatrix},$$

where $w_{k,n}^j$ denotes the j th symbol in the part of file W_k that is stored at node- n in the systematic storage coding scheme, which is described in more detail below.

Storage: We use systematic encoding for the (N, K_C) -MDS storage code, as presented in Table II.⁴ The first K_C nodes are systematic nodes which store independent pieces of the files. The remaining $N - K_C$ nodes are parity nodes which store linear combinations of the symbols at systematic nodes. In Table II, for $n \in [K_C + 1 : N]$, $LC^n(\cdot)$ denotes the linear combination at the parity node n , the input of which are $w_{k,[1:K_C]}^j = \{w_{k,1}^j, \dots, w_{k,K_C}^j\}$. The vector stored at node n is denoted by D_n .

Randomness: The randomness consists of K_C^2 uniformly random and independent symbols from $\mathbb{F}_q$, denoted by

$$S = \begin{bmatrix} S_{1,1} & S_{1,2} & \dots & S_{1,K_C} \\ \vdots & \vdots & \ddots & \vdots \\ S_{K_C,1} & S_{K_C,1} & \dots & S_{K_C,K_C} \end{bmatrix},$$

which are independent of the database and unavailable to the user. The nodes $[1 : K_C]$ store the rows $[1 : K_C]$ of S

⁴The assumption of systematic encoding can be made without loss of generality. If the implementation of the MDS storage code is more general as in [8], the scheme can treat the coded data at the first K_C nodes as the systematic part and the remaining design of the scheme follows. After decoding, the user can solve the full-rank linear system to obtain W_k . User-privacy and database-privacy are still guaranteed with the same reasoning.

	node 1 D_1	node 2 D_2	...	node K_C D_{K_C}	node $K_C + 1$ D_{K_C+1}	...	node N D_N
W_1	$w_{1,1}^1$	$w_{1,2}^1$	...	w_{1,K_C}^1	$LC^{K_C+1}(w_{1,[1:K_C]}^1)$	...	$LC^N(w_{1,[1:K_C]}^1)$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$w_{1,1}^{N-K_C}$	$w_{1,2}^{N-K_C}$	...	$w_{1,K_C}^{N-K_C}$	$LC^{K_C+1}(w_{1,[1:K_C]}^{N-K_C})$	...	$LC^N(w_{1,[1:K_C]}^{N-K_C})$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
W_K	$w_{K,1}^1$	$w_{K,2}^1$	...	w_{K,K_C}^1	$LC^{K_C+1}(w_{K,[1:K_C]}^1)$	...	$LC^N(w_{K,[1:K_C]}^1)$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$w_{K,1}^{N-K_C}$	$w_{K,2}^{N-K_C}$	...	$w_{K,K_C}^{N-K_C}$	$LC^{K_C+1}(w_{K,[1:K_C]}^{N-K_C})$	...	$LC^N(w_{K,[1:K_C]}^{N-K_C})$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

TABLE II
SYSTEMATIC (N, K_C) -MDS STORAGE CODE.

respectively. The randomness stored at nodes $[K_C + 1 : N]$ is generated using the same (N, K_C) -MDS storage code as used above for the files.

For the details of the queries and answers, w.l.o.g., assume the desired file is W_1 . The user privately generates K_C uniformly random vectors $U_1, \dots, U_{K_C}$ of length $(N - K_C)K$ over $\mathbb{F}_q$. The detailed achievable scheme is presented in two orthogonal cases as follows.

- **Case 1** ($N - K_C \leq K_C$)

Queries: The query to each node consists of K_C vectors over $\mathbb{F}_q$ as shown in Table III. Recall that e_i denotes the unit vector with a one at the i th entry, and zeros at all other entries. Specifically, for systematic nodes, $N - K_C$ out of the K_C query vectors retrieve the $N - K_C$ symbols of W_1 stored at each node by adding the unit vectors, in a shifted way among all systematic nodes. The queries to the parity nodes are just the K_C random vectors $U_1, \dots, U_{K_C}$. It can be observed that each node receives statistically uniformly random query vectors. Hence, user-privacy is guaranteed.

Answers: Each node receives K_C query vectors, and for each forms the inner product with the stored data vector, resulting in K_C symbols. Next, they add linear combinations of the random symbols $S_{i,j}$ to the resulting K_C symbols according to the storage code construction, and send the results to the user. Let $X_{i,j} = U_j D_i + S_{i,j}$, where $i \in [1 : K_C]$ denotes the index of systematic nodes and $j \in [1 : K_C]$ denotes the index of query vectors for each node, there are K_C^2 unknowns generated from queries, stored data, and randomness as follows,

$$\begin{aligned}
 & \begin{bmatrix} X_{1,1} & \dots & X_{1,K_C} \\ \vdots & \ddots & \vdots \\ X_{K_C,1} & \dots & X_{K_C,K_C} \end{bmatrix} \\
 &= \begin{bmatrix} U_1 D_1 + S_{1,1} & \dots & U_{K_C} D_1 + S_{1,K_C} \\ \vdots & \ddots & \vdots \\ U_1 D_{K_C} + S_{K_C,1} & \dots & U_{K_C} D_{K_C} + S_{K_C,K_C} \end{bmatrix}. \quad (15)
 \end{aligned}$$

The user receives K_C answers from each node, as shown in Table IV. Note that there are NK_C unknowns, among which K_C^2 unknowns are the $X_{i,j}$'s and $(N - K_C)K_C$ unknowns are symbols of the requested file. It can be observed that there are NK_C linearly

independent equations. Hence, the linear system is solvable. Besides the desired file W_k , the user obtains K_C^2 random variables $\{X_{i,j}\}_{i \in [1:K_C], j \in [1:K_C]}$. To show that database-privacy is guaranteed, we need to show that $I(W_k; \{X_{i,j}\}_{i \in [1:K_C], j \in [1:K_C]} | W_k, U_{[1:K_C]}) = 0$, which holds because

$$\begin{aligned}
 & I(W_k; \{X_{i,j}\}_{i \in [1:K_C], j \in [1:K_C]} | W_k, U_{[1:K_C]}) \\
 &= H(\{X_{i,j}\}_{i \in [1:K_C], j \in [1:K_C]} | W_k, U_{[1:K_C]}) \\
 &\quad - H(\{X_{i,j}\}_{i \in [1:K_C], j \in [1:K_C]} | W_{[1:K_C]}, U_{[1:K_C]}) \\
 &= K_C^2 L \log q - K_C^2 L \log q = 0.
 \end{aligned}$$

- **Case 2** ($N - K_C > K_C$)

Queries: Let $\beta = N - K_C \pmod{K_C}$, and $N - K_C = \alpha K_C + \beta$. The queries are as shown in Table V. Specifically, for systematic nodes, β out of the K_C query vectors retrieve the *first* β symbols of the requested file, in a shifted way among all systematic nodes. The remaining αK_C symbols at each systematic node are retrieved at the parity nodes. Since every K_C symbols need K_C independent linear equations, they are retrieved at K_C parity nodes. Similar as in Case 1, user-privacy is guaranteed because nodes receive statistically uniformly random query vectors.

Answers: The answers are generated in the same way as in Case 1, that is, by forming inner products of the received query vectors and the stored data vectors, and then adding randomness symbols. With $X_{i,j}$'s defined the same way as in (15), the answers received by the user is summarized in Table VI. Similarly as in Case 1, the linear system is solvable, hence W_1 can be decoded. Besides, database-privacy is guaranteed by the randomness.

VI. TSPIR

In this section, we discuss the SPIR problem with T -colluding servers, where there is no coding in the storage such that the database is simply replicated at all the storage nodes. The capacity of TSPIR is presented in Corollary 2.

A. Converse

1) *The proof for $R \leq C_{\text{TSPIR}}$:* From user-privacy (5), for any T nodes $\mathcal{T} \subset [1 : N]$, $H(A_{\mathcal{T}}^{[k]}, Q_{\mathcal{T}}^{[k]}) = H(A_{\mathcal{T}}^{[k']}, Q_{\mathcal{T}}^{[k']})$ and $H(Q_{\mathcal{T}}^{[k]}) = H(Q_{\mathcal{T}}^{[k']})$. Taking subtraction on both sides of

$Q_1^1 :$	$Q_2^1 :$	$\dots$	$Q_{K_C}^1 :$	$Q_{K_C+1}^1 :$	$\dots$	$Q_N^1 :$
$U_1 + e_1$	U_1		$U_1 + e_2$	U_1		U_1
$U_2 + e_2$	$U_2 + e_1$		$U_2 + e_3$	U_2		U_2
$\vdots$	$\vdots$		$\vdots$	$\vdots$		$\vdots$
$U_{N-K_C-1} + e_{N-K_C-1}$	$U_{N-K_C-1} + e_{N-K_C-2}$		$U_{N-K_C-1} + e_{N-K_C}$	U_{N-K_C-1}		U_{N-K_C-1}
$U_{N-K_C} + e_{N-K_C}$	$U_{N-K_C} + e_{N-K_C-1}$		U_{N-K_C}	U_{N-K_C}		U_{N-K_C}
U_{N-K_C+1}	$U_{N-K_C+1} + e_{N-K_C}$		U_{N-K_C+1}	U_{N-K_C+1}		U_{N-K_C+1}
U_{N-K_C+2}	U_{N-K_C+2}		U_{N-K_C+2}	U_{N-K_C+2}		U_{N-K_C+2}
$\vdots$	$\vdots$		$\vdots$	$\vdots$		$\vdots$
U_{K_C-1}	U_{K_C-1}		U_{K_C-1}	U_{K_C-1}		U_{K_C-1}
U_{K_C}	U_{K_C}		$U_{K_C} + e_1$	U_{K_C}		U_{K_C}

TABLE III
QUERIES WHEN USER WANTS W_1 AND $N - K_C \leq K_C$.

$A_1^1 :$	$A_2^1 :$	$\dots$	$A_{K_C}^1 :$	$A_{K_C+1}^1 :$	$\dots$	$A_N^1 :$
$X_{1,1} + w_{1,1}^1$	$X_{2,1}$		$X_{K_C,1} + w_{1,K_C}^1$	$LC^{K_C+1}(X_{[1:K_C],1})$		$LC^N(X_{[1:K_C],1})$
$X_{1,2} + w_{1,1}^1$	$X_{2,2} + w_{1,2}^1$		$X_{K_C,2} + w_{1,K_C}^1$	$LC^{K_C+1}(X_{[1:K_C],2})$		$LC^N(X_{[1:K_C],2})$
$\vdots$	$\vdots$		$\vdots$	$\vdots$		$\vdots$
$X_{1,N-K_C-1} + w_{1,1}^{N-K_C-1}$	$X_{2,N-K_C-1} + w_{1,2}^{N-K_C-2}$		$X_{K_C,N-K_C-1} + w_{1,K_C}^{N-K_C}$	$LC^{K_C+1}(X_{[1:K_C],N-K_C-1})$		$LC^N(X_{[1:K_C],N-K_C-1})$
$X_{1,N-K_C} + w_{1,1}^{N-K_C}$	$X_{2,N-K_C} + w_{1,2}^{N-K_C-1}$		$X_{K_C,N-K_C}$	$LC^{K_C+1}(X_{[1:K_C],N-K_C})$		$LC^N(X_{[1:K_C],N-K_C})$
$X_{1,N-K_C+1}$	$X_{2,N-K_C+1} + w_{1,2}^{N-K_C}$		$X_{K_C,N-K_C+1}$	$LC^{K_C+1}(X_{[1:K_C],N-K_C+1})$		$LC^N(X_{[1:K_C],N-K_C+1})$
$X_{1,N-K_C+2}$	$X_{2,N-K_C+2}$		$X_{K_C,N-K_C+2}$	$LC^{K_C+1}(X_{[1:K_C],N-K_C+2})$		$LC^N(X_{[1:K_C],N-K_C+2})$
$\vdots$	$\vdots$		$\vdots$	$\vdots$		$\vdots$
X_{1,K_C-1}	X_{2,K_C-1}		X_{K_C,K_C-1}	$LC^{K_C+1}(X_{[1:K_C],K_C-1})$		$LC^N(X_{[1:K_C],K_C-1})$
X_{1,K_C}	X_{2,K_C}		$X_{K_C,K_C} + w_{1,K_C}^1$	$LC^{K_C+1}(X_{[1:K_C],K_C})$		$LC^N(X_{[1:K_C],K_C})$

TABLE IV
ANSWERS RECEIVED BY USER WHEN W_1 IS DESIRED AND $N - K_C \leq K_C$.

$Q_1^1 :$	$Q_2^1 :$	$\dots$	$Q_{K_C}^1 :$	$Q_{K_C+1}^1 \sim Q_{2K_C}^1 :$	$Q_{2K_C+1}^1 \sim Q_{3K_C}^1 :$	$\dots$	$Q_{\alpha K_C+1}^1 \sim Q_{(\alpha+1)K_C}^1 :$	$Q_{(\alpha+1)K_C+1}^1 \sim Q_N^1 :$
$U_1 + e_1$	U_1		$U_1 + e_2$	$U_1 + e_{\beta+1}$	$U_1 + e_{K_C+\beta+1}$		$U_1 + e_{(\alpha-1)K_C+\beta+1}$	U_1
$U_2 + e_2$	$U_2 + e_1$		$U_2 + e_3$	$U_2 + e_{\beta+2}$	$U_2 + e_{K_C+\beta+2}$		$U_2 + e_{(\alpha-1)K_C+\beta+2}$	U_2
$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
$U_{\beta-1} + e_{\beta-1}$	$U_{\beta-1} + e_{\beta-2}$		$U_{\beta-1} + e_{\beta}$	$U_{\beta-1} + e_{2\beta-1}$	$U_{\beta-1} + e_{K_C+2\beta-1}$		$U_{\beta-1} + e_{(\alpha-1)K_C+2\beta-1}$	$U_{\beta-1}$
$U_{\beta} + e_{\beta}$	$U_{\beta} + e_{\beta-1}$		U_{β}	$U_{\beta} + e_{2\beta}$	$U_{\beta} + e_{K_C+2\beta}$		$U_{\beta} + e_{(\alpha-1)K_C+2\beta}$	U_{β}
$U_{\beta+1}$	$U_{\beta+1} + e_{\beta}$		$U_{\beta+1}$	$U_{\beta+1} + e_{2\beta+1}$	$U_{\beta+1} + e_{K_C+2\beta+1}$		$U_{\beta+1} + e_{(\alpha-1)K_C+2\beta+1}$	$U_{\beta+1}$
$U_{\beta+2}$	$U_{\beta+2}$		$U_{\beta+2}$	$U_{\beta+2} + e_{2\beta+2}$	$U_{\beta+2} + e_{K_C+2\beta+2}$		$U_{\beta+2} + e_{(\alpha-1)K_C+2\beta+2}$	$U_{\beta+2}$
$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
U_{K_C-1}	U_{K_C-1}		U_{K_C-1}	$U_{K_C-1} + e_{\beta+K_C-1}$	$U_{K_C-1} + e_{2K_C+\beta-1}$		$U_{K_C-1} + e_{\alpha K_C+\beta-1}$	U_{K_C-1}
U_{K_C}	U_{K_C}		$U_{K_C} + e_1$	$U_{K_C} + e_{\beta+K_C}$	$U_{K_C} + e_{2K_C+\beta}$		$U_{K_C} + e_{\alpha K_C+\beta}$	U_{K_C}

TABLE V
QUERIES WHEN USER WANTS W_1 AND $N - K_C > K_C$.

the equations, we have that $H(A_{\mathcal{T}}^{[k]}|Q_{\mathcal{T}}^{[k]}) = H(A_{\mathcal{T}}^{[k']}|Q_{\mathcal{T}}^{[k']})$. the detailed proof. Similarly, $H(A_{\mathcal{T}}^{[k]}|W_k, Q_{\mathcal{T}}^{[k]}) = H(A_{\mathcal{T}}^{[k']}|W_k, Q_{\mathcal{T}}^{[k']})$.

$$\begin{aligned}
H(W_k) &= H(W_k|\mathcal{Q}) - H(W_k|A_{[1:N]}^{[k]}, \mathcal{Q}) \\
&= H(A_{[1:N]}^{[k]}|\mathcal{Q}) - H(A_{[1:N]}^{[k]}|W_k, \mathcal{Q}) \\
&\leq H(A_{[1:N]}^{[k]}|\mathcal{Q}) - H(A_{\mathcal{T}}^{[k]}|W_k, \mathcal{Q}) \\
&= H(A_{[1:N]}^{[k]}|\mathcal{Q}) - H(A_{\mathcal{T}}^{[k]}|W_k, Q_{\mathcal{T}}^{[k]}) \\
&= H(A_{[1:N]}^{[k]}|\mathcal{Q}) - H(A_{\mathcal{T}}^{[k']}|W_k, Q_{\mathcal{T}}^{[k']}) \\
&= H(A_{[1:N]}^{[k]}|\mathcal{Q}) - H(A_{\mathcal{T}}^{[k']}|Q_{\mathcal{T}}^{[k']}) \\
&= H(A_{[1:N]}^{[k]}|\mathcal{Q}) - H(A_{\mathcal{T}}^{[k]}|Q_{\mathcal{T}}^{[k]}) \\
&= H(A_{[1:N]}^{[k]}|\mathcal{Q}) - H(A_{\mathcal{T}}^{[k]}|\mathcal{Q}).
\end{aligned}$$

By averaging over all $\mathcal{T}$ and using Han's inequality, with similar steps as in Section V-A1, it can be shown that $R = \frac{H(W_k)}{\sum_{n=1}^N H(A_n^{[k]})} \leq 1 - \frac{T}{N}$.

The proof of $\rho \geq \frac{T}{N-T}$ follows similarly as in Section V-A2 by replacing $n_{[1:K_C]}$ with $\mathcal{T}$ and K_C with T . Hence we omit

B. Achievability

Assume each file consists of $L = N - T$ symbols from a large enough field $\mathbb{F}_q$, denoted by $W_k = \{W_{k,1}, W_{k,2}, \dots, W_{k,N-T}\}$. The database which is stored at each node is represented by a length- $K(N - T)$ vector $D = [W_{1,1}, \dots, W_{1,N-T}, \dots, W_{K,1}, \dots, W_{K,N-T}]$. A deterministic generating matrix of an (N, T) -GRS code, denoted as $\mathbf{G}_{(N,T)}$, is publicly known to the user and all nodes. The user wants to retrieve W_k privately.

The queries are generated in the following way. The user privately generates T uniformly i.i.d. random vectors $U_1, \dots, U_T$ of length $K(N - T)$ over $\mathbb{F}_q$. Let $e_{k,i}$ denote the length- $K(N - T)$ unit vector where only the $((k-1)(N-T) + i)$ th entry is 1 and all the other entries are 0's. The purpose of $e_{k,i}$ is to retrieve the i th symbol of W_k . The query vectors are generated by

$$[Q_1^{[k]}, \dots, Q_N^{[k]}] = [U_1, \dots, U_T] \cdot \mathbf{G}_{(N,T)}$$

$A_1^1 :$	$X_{1,1} + w_{1,1}^1, X_{1,2} + w_{1,1}^2, \dots, X_{1,\beta-1} + w_{1,1}^{\beta-1}, X_{1,\beta} + w_{1,1}^\beta, X_{1,\beta+1}, X_{1,\beta+2}, \dots, X_{1,K_C-1}, X_{1,K_C}$		
$A_2^1 :$	$X_{2,1}, X_{2,2} + w_{1,2}^1, \dots, X_{2,\beta-1} + w_{1,2}^{\beta-1}, X_{2,\beta} + w_{1,2}^\beta, X_{2,\beta+1} + w_{1,2}^\beta, X_{2,\beta+2}, \dots, X_{2,K_C-1}, X_{2,K_C}$		
$\vdots$	$\vdots$		
$A_{K_C}^1 :$	$X_{K_C,1} + w_{1,K_C}^2, X_{K_C,2} + w_{1,K_C}^3, \dots, X_{K_C,\beta-1} + w_{1,K_C}^\beta, X_{K_C,\beta}, X_{K_C,\beta+1}, X_{K_C,\beta+2}, \dots, X_{K_C,K_C-1}, X_{K_C,K_C} + w_{1,K_C}^1$		
$A_{K_C+1}^1 :$	$LC^{K_C+1}(X_{[1:K_C],1}) + LC^{K_C+1}(w_{1,[1:K_C]}^{\beta+1})$	$\dots$	$LC^{K_C+1}(X_{[1:K_C],K_C}) + LC^{K_C+1}(w_{1,[1:K_C]}^{\beta+K_C})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$A_{2K_C}^1 :$	$LC^{2K_C}(X_{[1:K_C],1}) + LC^{2K_C}(w_{1,[1:K_C]}^{\beta+1})$	$\dots$	$LC^{2K_C}(X_{[1:K_C],K_C}) + LC^{2K_C}(w_{1,[1:K_C]}^{\beta+K_C})$
$A_{2K_C+1}^1 :$	$LC^{2K_C+1}(X_{[1:K_C],1}) + LC^{2K_C+1}(w_{1,[1:K_C]}^{K_C+\beta+1})$	$\dots$	$LC^{2K_C+1}(X_{[1:K_C],K_C}) + LC^{2K_C+1}(w_{1,[1:K_C]}^{2K_C+\beta})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$A_{3K_C}^1 :$	$LC^{3K_C}(X_{[1:K_C],1}) + LC^{3K_C}(w_{1,[1:K_C]}^{K_C+\beta+1})$	$\dots$	$LC^{3K_C}(X_{[1:K_C],K_C}) + LC^{3K_C}(w_{1,[1:K_C]}^{2K_C+\beta})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$A_{\alpha K_C+1}^1 :$	$LC^{\alpha K_C+1}(X_{[1:K_C],1}) + LC^{\alpha K_C+1}(w_{1,[1:K_C]}^{(\alpha-1)K_C+\beta+1})$	$\dots$	$LC^{\alpha K_C+1}(X_{[1:K_C],K_C}) + LC^{\alpha K_C+1}(w_{1,[1:K_C]}^{\alpha K_C+\beta})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$A_{(\alpha+1)K_C}^1 :$	$LC^{(\alpha+1)K_C}(X_{[1:K_C],1}) + LC^{(\alpha+1)K_C}(w_{1,[1:K_C]}^{(\alpha-1)K_C+\beta+1})$	$\dots$	$LC^{(\alpha+1)K_C}(X_{[1:K_C],K_C}) + LC^{(\alpha+1)K_C}(w_{1,[1:K_C]}^{\alpha K_C+\beta})$
$A_{(\alpha+1)K_C+1}^1 :$	$LC^{(\alpha+1)K_C+1}(X_{[1:K_C],1})$	$\dots$	$LC^{(\alpha+1)K_C+1}(X_{[1:K_C],K_C})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$A_N^1 :$	$LC^N(X_{[1:K_C],1})$	$\dots$	$LC^N(X_{[1:K_C],K_C})$

TABLE VI

ANSWERS RECEIVED BY USER WHEN W_1 IS DESIRED AND $N - K_C > K_C$.

$$+ [\underbrace{\vec{0}, \dots, \vec{0}}_{T \text{ zero vectors}}, e_{k,1}, \dots, e_{k,N-T}]. \quad (16) \quad \text{with size } |\mathcal{N}| = K_C + T - 1,$$

To guarantee database-privacy from the user, the nodes share T uniformly i.i.d. symbols $[S_1, \dots, S_T] = S$, which is the randomness and unavailable to the user. The servers generate their answers by taking the inner product of the query vector and the stored data vector, then add on a linear combination of the randomness in the following way,

$$A_n^{[k]} = \langle Q_n^{[k]}, D \rangle + \langle \mathbf{G}_{(N,T)}(n), S \rangle, \quad (17)$$

where $\mathbf{G}_{(N,T)}(n)$ denotes the n th column of matrix $\mathbf{G}_{(N,T)}$. Let $X_j = \langle U_j, D \rangle + S_j$, where $j = 1, \dots, T$, the answers received by the user are

$$[A_1^{[k]}, \dots, A_N^{[k]}] = [X_1, \dots, X_T, W_{k,1}, \dots, W_{k,N-T}] \cdot \begin{bmatrix} \mathbf{G}_{(N,T)} \\ \mathbf{0} \quad \mathbf{I} \end{bmatrix}, \quad (18)$$

where we omit the dimension of the zero matrix $\mathbf{0}$ and the identity matrix $\mathbf{I}$ because there is no ambiguity. Because $\mathbf{G}_{(N,T)}$ is the generating matrix of an (N, T) -GRS code, the matrix $\begin{bmatrix} \mathbf{G}_{(N,T)} \\ \mathbf{0} \quad \mathbf{I} \end{bmatrix}$ is invertible. Therefore, the user can solve the linear system (18), hence obtain W_k .

It is direct to check that both user-privacy and database-privacy are guaranteed. The scheme achieves a rate $R = 1 - \frac{T}{N}$, with the ratio of total randomness size to file size being $\rho = \frac{T}{N-T}$.

VII. MDS-TSPIR WITH ADDITIVE RANDOMNESS

A. Converse

In this section, we show the converse part of Theorem 3 on the capacity of MDS-TSPIR schemes with additive randomness, as defined in (2).

Lemma 8. *For any achievable MDS-TSPIR scheme with additive randomness, and for any set of nodes $\mathcal{N} \subset [1 : N]$*

$$H(A_{\mathcal{N}}^{[k]} | W_k, Q_{\mathcal{N}}^{[k]}) = H(A_{\mathcal{N}}^{[k]} | Q_{\mathcal{N}}^{[k]}). \quad (19)$$

Proof: W.l.o.g., assume the $K_C + T - 1$ nodes are $\mathcal{N} = [1 : K_C + T - 1]$.

Case 1 ($K_S \geq K_C + T - 1$): If the dimension of the MDS code for the randomness $K_S \geq K_C + T - 1$, i.e., $S_{[1:K_C+T-1]}$ are linearly and statistically independent, (19) holds. Because from database-privacy, $H(A_{\mathcal{N}}^{[k]} | W_k, Q_{\mathcal{N}}^{[k]}) = H(A_{\mathcal{N}}^{[k]} | W_{[1:K]}, Q_{\mathcal{N}}^{[k]}) = H(S_{[1:K_C+T-1]}) = (K_C + T - 1)H(S_1)$; and from the additive randomness requirement (2), $H(A_{\mathcal{N}}^{[k]} | Q_{\mathcal{N}}^{[k]}) \leq (K_C + T - 1)H(A_1^{[k]} | Q_1^{[k]}) = (K_C + T - 1)H(S_1)$. On the other hand, $H(A_{\mathcal{N}}^{[k]} | Q_{\mathcal{N}}^{[k]}) \geq H(A_{\mathcal{N}}^{[k]} | W_k, Q_{\mathcal{N}}^{[k]})$. Hence, (19) holds.

Case 2 ($K_S < K_C + T - 1$): If $K_S < K_C + T - 1$, $S_{[1:K_C+T-1]}$ has linear dependency. Thus, there exist nullity($S_{\mathcal{N}})$ = $K_C + T - 1 - K_S$ set of coefficients $\{\alpha_n, n \in [1 : K_C + T - 1]\}$, where in each set the coefficients are not all zeros, such that $\sum_{n=1}^{K_C+T-1} \alpha_n S_n = 0$. From (2), $\sum_{n=1}^{K_C+T-1} \alpha_n A_n^{[k]} = \sum_{n=1}^{K_C+T-1} \alpha_n F_n^{[k]}(Q_n^{[k]}, D_n)$, which does not contain the randomness. If $\sum_{n=1}^{K_C+T-1} \alpha_n F_n^{[k]} = 0$, $I(\sum_{n=1}^{K_C+T-1} \alpha_n F_n^{[k]}; W_k | Q_{[1:K_C+T-1]}^{[k]}) = 0$ holds. Otherwise, $\sum_{n=1}^{K_C+T-1} \alpha_n F_n^{[k]}$ is a nonzero function of W_k . (By database-privacy, $\sum_{n=1}^{K_C+T-1} \alpha_n F_n^{[k]}$ is independent of W_k .)

Among the $K_C + T - 1$ nodes, any T nodes may collude. Let $W_k = (D_1^k, D_2^k, \dots, D_{K_C}^k)$, where D_n^k denote the information about file W_k stored at node n . The function $\sum_{n=1}^{K_C+T-1} \alpha_n F_n^{[k]} = \sum_{n=1}^{K_C-1} \alpha_n F_n^{[k]} + \sum_{n=K_C}^{K_C+T-1} \alpha_n F_n^{[k]}$ depends on $D_{K_C}^k$ if and only if $\sum_{n=K_C}^{K_C+T-1} \alpha_n F_n^{[k]}$ depends on $D_{K_C}^k$, because $F_{[1:K_C-1]}^{[k]}$ are functions of the stored data $D_{[1:K_C-1]}^k$, which are independent of $D_{K_C}^k$. However, if $\sum_{n=K_C}^{K_C+T-1} \alpha_n F_n^{[k]}$ depends on $D_{K_C}^k$, the T nodes $[K_C : K_C + T - 1]$ can collude and know the desired file is W_k , because by database-privacy $\sum_{n=K_C}^{K_C+T-1} \alpha_n F_n^{[k]}$ is independent of $D_{K_C}^{k'}$.

for any $k' \neq k$. (The T nodes know the codes for the randomness, hence know the coefficients $\{\alpha_n\}$.) Similarly, for any $n' \in [1 : K_C - 1]$, the T nodes $\{n'\} \cup [K_C + 1 : K_C + T - 1]$ can collude. Therefore, $\sum_{n=1}^{K_C+T-1} \alpha_n F_n^{[k]}$ should be independent of D_n^k , for all $n' \in [1 : K_C - 1]$. That is, for any $n \in [1 : K_C]$,

$$I\left(\sum_{n=1}^{K_C+T-1} \alpha_n F_n^{[k]}; D_n^k | Q_{[1:K_C+T-1]}^{[k]}\right) = 0. \quad (20)$$

Because when the randomness is canceled, the user cannot learn any information about $W_{\bar{k}}$, the sum of answers is independent of $W_{\bar{k}}$, conditioned on W_k . Therefore,

$$\begin{aligned} & I\left(\sum_{n=1}^{K_C+T-1} \alpha_n F_n^{[k]}; D_{K_C}^k | D_{[1:K_C-1]}^k, Q_{[1:K_C+T-1]}^{[k]}\right) \\ &= I\left(\sum_{n=1}^{K_C+T-1} \alpha_n F_n^{[k]}; D_{K_C}^k | D_{[1:K_C-1]}^k, Q_{[1:K_C+T-1]}^{[k]}\right) \\ &= H(D_{K_C}^k | D_{[1:K_C-1]}^k, Q_{[1:K_C+T-1]}^{[k]}) \\ &\quad - H(D_{K_C}^k | \sum_{n=1}^{K_C+T-1} \alpha_n F_n^{[k]}, D_{[1:K_C-1]}^k, Q_{[1:K_C+T-1]}^{[k]}) \\ &\stackrel{(a)}{=} H(D_{K_C}^k | D_{[1:K_C-1]}^k, Q_{[1:K_C+T-1]}^{[k]}) \\ &\quad - H(D_{K_C}^k | \sum_{n=K_C}^{K_C+T-1} \alpha_n F_n^{[k]}, D_{[1:K_C-1]}^k, Q_{[1:K_C+T-1]}^{[k]}) \\ &= I(D_{K_C}^k; \sum_{n=K_C}^{K_C+T-1} \alpha_n F_n^{[k]} | D_{[1:K_C-1]}^k, Q_{[1:K_C+T-1]}^{[k]}), \end{aligned}$$

where (a) holds because $\sum_{n=1}^{K_C-1} \alpha_n F_n^{[k]}$ is a deterministic function of $D_{[1:K_C-1]}^k$ and $Q_{[1:K_C-1]}^{[k]}$.

From database-privacy, for any $k' \neq k$, $I(D_{K_C}^{k'}; \sum_{n=K_C}^{K_C+T-1} \alpha_n F_n^{[k]} | D_{[1:K_C-1]}^k, Q_{[1:K_C+T-1]}^{[k]}) = 0$. If $I(D_{K_C}^k; \sum_{n=K_C}^{K_C+T-1} \alpha_n F_n^{[k]} | D_{[1:K_C-1]}^k, Q_{[1:K_C+T-1]}^{[k]}) \neq 0$, the T nodes $[K_C : K_C + T - 1]$ can collude and learn the desired file index k . Therefore, $I(\sum_{n=1}^{K_C+T-1} \alpha_n F_n^{[k]}; D_{K_C}^k | D_{[1:K_C-1]}^k, Q_{[1:K_C+T-1]}^{[k]}) = 0$. For any node set $\mathcal{M}' \subset [1 : K_C - 1]$,

$$\begin{aligned} & I\left(\sum_{n=1}^{K_C+T-1} \alpha_n F_n^{[k]}; D_{K_C}^k | D_{\mathcal{M}'}^k, Q_{[1:K_C+T-1]}^{[k]}\right) \\ &\leq I\left(\sum_{n=1}^{K_C+T-1} \alpha_n F_n^{[k]}; D_{K_C}^k | D_{[1:K_C-1]}^k, Q_{[1:K_C+T-1]}^{[k]}\right) = 0. \end{aligned} \quad (21)$$

Combining (20) and (21), and noting that $W_k = D_{[1:K_C]}^k$, for all $K_C + T - 1 - K_S$ set of coordinates $\{\alpha_n, n \in [1 : K_C + T - 1]\}$ such that $\sum_{n=1}^{K_C+T-1} \alpha_n S_n = 0$,

$$I\left(\sum_{n=1}^{K_C+T-1} \alpha_n F_n^{[k]}; W_k | Q_{[1:K_C+T-1]}^{[k]}\right) = 0. \quad (22)$$

To show that (19) holds, we note that

$$I(A_{[1:K_C+T-1]}^{[k]}; W_k | Q_{[1:K_C+T-1]}^{[k]})$$

$$\begin{aligned} &= H(A_{[1:K_C+T-1]}^{[k]} | Q_{[1:K_C+T-1]}^{[k]}) \\ &\quad - H(A_{[1:K_C+T-1]}^{[k]} | W_k, Q_{[1:K_C+T-1]}^{[k]}) \\ &= H(A_{[1:K_C+T-1]}^{[k]} | W_{\bar{k}}, Q_{[1:K_C+T-1]}^{[k]}) \\ &\quad - H(A_{[1:K_C+T-1]}^{[k]} | W_{[1:K]}, Q_{[1:K_C+T-1]}^{[k]}) \\ &= H(A_{[1:K_C+T-1]}^{[k]} | W_{\bar{k}}, Q_{[1:K_C+T-1]}^{[k]}) - H(S_{[1:K_C+T-1]}) \\ &= H(\{F_n^{[k]} + S_n\}_{n \in [1:K_C+T-1]} | W_{\bar{k}}, Q_{[1:K_C+T-1]}^{[k]}) - K_S H(S_1). \end{aligned}$$

If $H(\{F_n^{[k]} + S_n\}_{n \in [1:K_C+T-1]} | W_{\bar{k}}, Q_{[1:K_C+T-1]}^{[k]}) > K_S H(S_1)$, the rank of $\{F_n^{[k]} + S_n\}_{n \in [1:K_C+T-1]}$ is at least $K_S + 1$, then the nullity of $\{F_n^{[k]} + S_n\}_{n \in [1:K_C+T-1]}$ is at most $K_C + T - 2 - K_S$. There exists at least one set of coefficients $\{\alpha_n\}$ such that $\sum_{n=1}^{K_C+T-1} \alpha_n S_n = 0$ but $\sum_{n=1}^{K_C+T-1} \alpha_n (F_n^{[k]} + S_n) \neq 0$. In this case, $\sum_{n=1}^{K_C+T-1} \alpha_n (F_n^{[k]} + S_n)$ depends on W_k , which contradicts with (22).

Therefore, $I(A_{[1:K_C+T-1]}^{[k]}; W_k | Q_{[1:K_C+T-1]}^{[k]}) = 0$ and (19) holds. $\square$

1) *The proof for $R \leq C_{MDS-TSPR}^{+S}$* : For any file W_k , $k \in [1 : K]$, and any set of nodes $\mathcal{N} \subset [1 : N]$ with size $K_C + T - 1$,

$$\begin{aligned} H(W_k) &= H(W_k | \mathcal{Q}) \\ &\stackrel{(a)}{=} H(W_k | \mathcal{Q}) - H(W_k | A_{[1:N]}^{[k]}, \mathcal{Q}) \\ &= H(A_{[1:N]}^{[k]} | \mathcal{Q}) - H(A_{[1:N]}^{[k]} | W_k, \mathcal{Q}) \\ &\leq H(A_{[1:N]}^{[k]} | \mathcal{Q}) - H(A_{\mathcal{N}}^{[k]} | W_k, \mathcal{Q}, Q_{\mathcal{N}}^{[k]}) \\ &\stackrel{(b)}{=} H(A_{[1:N]}^{[k]} | \mathcal{Q}) - H(A_{\mathcal{N}}^{[k]} | W_k, Q_{\mathcal{N}}^{[k]}) \\ &\stackrel{(c)}{=} H(A_{[1:N]}^{[k]} | \mathcal{Q}) - H(A_{\mathcal{N}}^{[k]} | Q_{\mathcal{N}}^{[k]}) \\ &\leq H(A_{[1:N]}^{[k]} | \mathcal{Q}) - H(A_{\mathcal{N}}^{[k]} | \mathcal{Q}), \end{aligned}$$

where (a) follows from (3). Equalities (c) follow from Lemma 8.

Averaging over all $\mathcal{N}$ with size $K_C + T - 1$, we have that

$$H(W_k) \leq H(A_{[1:N]}^{[k]} | \mathcal{Q}) - \frac{1}{\binom{N}{K_C+T-1}} \sum_{\substack{\mathcal{N} \subset [1:N] \\ |\mathcal{N}|=K_C+T-1}} H(A_{\mathcal{N}}^{[k]} | \mathcal{Q}).$$

By Han's inequality [17],

$$\frac{1}{\binom{N}{K_C+T-1}} \sum_{\substack{\mathcal{N} \subset [1:N] \\ |\mathcal{N}|=K_C+T-1}} H(A_{\mathcal{N}}^{[k]} | \mathcal{Q}) \geq \frac{K_C + T - 1}{N} H(A_{[1:N]}^{[k]} | \mathcal{Q}).$$

Therefore, $R = \frac{H(W_k)}{\sum_{n=1}^N H(A_n^{[k]})} \leq \frac{H(W_k)}{H(A_{[1:N]}^{[k]} | \mathcal{Q})} \leq 1 - \frac{K_C+T-1}{N}$. $\square$

2) *The proof for $\rho \geq \frac{K_C+T-1}{N-K_C-T+1}$* : Let $\mathcal{N}$ be any set of nodes with size $K_C + T - 1$, by database-privacy (6),

$$\begin{aligned} 0 &= I(W_{\bar{k}}; A_{[1:N]}^{[k]} | \mathcal{Q}) \\ &= H(W_{\bar{k}} | \mathcal{Q}) - H(W_{\bar{k}} | A_{[1:N]}^{[k]}, \mathcal{Q}) \\ &\stackrel{(a)}{=} H(W_{\bar{k}} | \mathcal{Q}, W_k) - H(W_{\bar{k}} | A_{[1:N]}^{[k]}, \mathcal{Q}, W_k) \\ &= I(W_{\bar{k}}; A_{[1:N]}^{[k]} | \mathcal{Q}, W_k) \end{aligned}$$

$$\begin{aligned}
&\geq I(W_k; A_{\mathcal{N}}^{[k]} | \mathcal{Q}, W_k) \\
&\stackrel{(b)}{=} H(A_{\mathcal{N}}^{[k]} | \mathcal{Q}, W_k) - H(A_{\mathcal{N}}^{[k]} | \mathcal{Q}, W_{[1:K]}) + H(A_{\mathcal{N}}^{[k]} | \mathcal{Q}, W_{[1:K]}, S) \\
&= H(A_{\mathcal{N}}^{[k]} | \mathcal{Q}, W_k) - I(S; A_{\mathcal{N}}^{[k]} | \mathcal{Q}, W_{[1:K]}) \\
&\geq H(A_{\mathcal{N}}^{[k]} | \mathcal{Q}, W_k, Q_{\mathcal{N}}^{[k]}) - H(S) \\
&\stackrel{(c)}{=} H(A_{\mathcal{N}}^{[k]} | Q_{\mathcal{N}}^{[k]}) - H(S) \\
&\geq H(A_{\mathcal{N}}^{[k]} | \mathcal{Q}) - H(S),
\end{aligned}$$

where (a) holds because W_k is independent of other files $W_{\bar{k}}$, and from $A_{[1:N]}^{[k]}$ and $\mathcal{Q}$ the user can decode W_k . Equality (b) holds because the answers $A_{\mathcal{N}}^{[k]}$ are deterministic functions of the queries $\mathcal{Q}$, the database $W_{[1:K]}$, and the randomness S . Equalities (c) follows from Lemma 8.

Averaging over all $\mathcal{N}$, and from the proof in Section VII-A1 above,

$$\begin{aligned}
H(S) &\geq \frac{1}{\binom{N}{K_C+T-1}} \sum_{\substack{\mathcal{N} \subseteq [1:N] \\ |\mathcal{N}|=K_C+T-1}} H(A_{\mathcal{N}}^{[k]} | \mathcal{Q}) \\
&\geq \frac{K_C + T - 1}{N} H(A_{[1:N]}^{[k]} | \mathcal{Q}) \\
&\geq \frac{K_C + T - 1}{N - K_C - T + 1} H(W_k).
\end{aligned}$$

$$\text{Hence, } \rho = \frac{H(S)}{H(W_k)} \geq \frac{K_C + T - 1}{N - K_C - T + 1}.$$

□

B. Achievability

In this section, we present a scheme which achieves the capacity with the lowest secrecy rate in Theorem 3. The storage code used in the scheme is an (N, K_C) generalized Reed Solomon (GRS) code. To achieve T -privacy, we generate queries by using an (N, T) -GRS code. The answers generated in this setting are codewords from the Schur-product of the storage code and query generating code. From [18], the minimum dimension of the Schur product of the two MDS codes is $K_C + T - 1$ (if $K_C + T - 1 \leq N$), achieved when the two MDS codes are generalized Reed Solomon codes with the same evaluation-point sequence. Indeed, the scheme in [15] for PIR with coded colluding databases uses GRS codes. Our scheme develops that in [15] to guarantee database-privacy.

Database: W.l.o.g., let $L = K_C(N - K_C - T + 1)$ be the file length, and let matrix $\mathbf{W}$ below represent all K files,

$$\mathbf{W} = \begin{bmatrix} w_{1,1}^{[1]} & w_{1,2}^{[1]} & \cdots & w_{1,K_C}^{[1]} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N-K_C-T+1,1}^{[1]} & w_{N-K_C-T+1,2}^{[1]} & \cdots & w_{N-K_C-T+1,K_C}^{[1]} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1,1}^{[K]} & w_{1,2}^{[K]} & \cdots & w_{1,K_C}^{[K]} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N-K_C-T+1,1}^{[K]} & w_{N-K_C-T+1,2}^{[K]} & \cdots & w_{N-K_C-T+1,K_C}^{[K]} \end{bmatrix},$$

where every $N - K_C - T + 1$ rows correspond to a file.

Storage: Let $\Lambda = (\lambda_1, \dots, \lambda_N) \in \mathbb{F}_q^N$ where $\lambda_i \neq \lambda_j$. Let $\Phi = (\phi_1, \dots, \phi_N) \in \mathbb{F}_q^N$ where $\phi_i \neq 0$. The generating matrix

of the storage code is

$$\mathbf{G}_S = \begin{bmatrix} 1 & \cdots & 1 \\ \lambda_1 & \cdots & \lambda_N \\ \vdots & \ddots & \vdots \\ \lambda_1^{K_C-1} & \cdots & \lambda_N^{K_C-1} \end{bmatrix} \cdot \text{diag}(\Phi).$$

The data stored at the N nodes are generated by

$$[D_1, \dots, D_N] = \mathbf{D} = \mathbf{W} \cdot \mathbf{G}_S.$$

Queries: Let $\Psi = (\psi_1, \dots, \psi_N) \in \mathbb{F}_q^N$ where $\psi_i \neq 0$. The generating matrix of queries is

$$\mathbf{G}_Q = \begin{bmatrix} 1 & \cdots & 1 \\ \lambda_1 & \cdots & \lambda_N \\ \vdots & \ddots & \vdots \\ \lambda_1^{T-1} & \cdots & \lambda_N^{T-1} \end{bmatrix} \cdot \text{diag}(\Psi).$$

The user queries consist of K_C sections. For each section r , $r \in [1 : K_C]$, the user generates T independent uniformly random vectors $U_1^{(r)}, \dots, U_T^{(r)}$ of length $(N - K_C - T + 1)K$ over $\mathbb{F}_q$. Denote $[U_1^{(r)}, \dots, U_T^{(r)}] = \mathbf{U}^{(r)}$, the N random vectors corresponding to each node are generated by

$$[\tilde{U}_1^{(r)}, \dots, \tilde{U}_N^{(r)}] = \tilde{\mathbf{U}}^{(r)} = \mathbf{U}^{(r)} \cdot \mathbf{G}_Q.$$

W.l.o.g., assume the desired file is W_1 . The detailed query scheme is presented in two orthogonal cases as follows.

- **Case 1** ($N - K_C - T + 1 \leq K_C$) Let e_i denote the unit vector of length $(N - K_C - T + 1)K$ with a one at the i th entry, and zeros at all other entries. Let matrix

$$\mathbf{E} = \begin{bmatrix} e_1 & 0 & \cdots & e_2 & 0 & \cdots & 0 \\ e_2 & e_1 & \cdots & e_3 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ e_{N-K_C-T} & e_{N-K_C-T-1} & \cdots & e_{N-K_C-T+1} & \cdots & \cdots & \cdots \\ e_{N-K_C-T+1} & e_{N-K_C-T} & \cdots & 0 & \cdots & \cdots & \cdots \\ 0 & e_{N-K_C-T+1} & \cdots & 0 & \cdots & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e_1 & 0 & \cdots & 0 \end{bmatrix}$$

be an $K_C(N - K_C - T + 1)K \times N$ matrix, where the unit vectors are designed in a shifted way among the first K_C nodes, to retrieve symbols of W_1 . The queries are generated by randomizing $\mathbf{E}$ from adding random matrices $\tilde{\mathbf{U}}^{(r)}$, i.e.,

$$Q_{[1:N]}^{[1]} = [\tilde{\mathbf{U}}^{(1)}, \dots, \tilde{\mathbf{U}}^{(K_C)}]^T + \mathbf{E}. \quad (23)$$

- **Case 2** ($N - K_C - T + 1 > K_C$) In this case, let $\beta = N - K_C - T + 1 \pmod{K_C}$, and $N - K_C - T + 1 = \alpha K_C + \beta$. The matrix $\mathbf{E}$ is as shown in Table VII, where E_n denotes the n th column of $\mathbf{E}$. The queries are generated by randomizing $\mathbf{E}$ in the same way as in (23).

For each node n , the query $Q_n^{[1]}$ is the n th column of $Q_{[1:N]}^{[1]}$, where every $(N - K_C - T + 1)K$ entries form a query vector for each section, denoted by $Q_n^{[1],(r)}$, $r \in [1 : K_C]$.

Answers: Each node receives K_C query vectors, and for each forms the inner product with the stored data vector, resulting in K_C symbols. To ensure database-privacy, let the randomness $\mathbf{S} = [S_j^{(r)}]_{j \in [1:K_C+T-1], r \in [1:K_C]}$, which is comprised of $K_C(K_C + T - 1)$ independent uniformly random symbols from $\mathbb{F}_q$. The randomness $\mathbf{S}$ is independent of the database and unavailable to the user. Denote $\phi_n \psi_n \sum_{j=1}^{K_C+T-1} S_j^{(r)} \cdot \lambda_n^{j-1}$

E_1	E_2	$\dots$	E_{K_C}	$E_{K_C+1} \sim E_{2K_C}$	$E_{2K_C+1} \sim E_{3K_C}$	$\dots$	$E_{\alpha K_C+1} \sim E_{(\alpha+1)K_C}$	$E_{(\alpha+1)K_C+1} \sim E_N$
e_1	0		e_2	$e_{\beta+1}$	$e_{K_C+\beta+1}$		$e_{(\alpha-1)K_C+\beta+1}$	0
e_2	e_1		e_3	$e_{\beta+2}$	$e_{K_C+\beta+2}$		$e_{(\alpha-1)K_C+\beta+2}$	0
$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
$e_{\beta-1}$	$e_{\beta-2}$	$\dots$	e_{β}	$e_{2\beta-1}$	$e_{K_C+2\beta-1}$	$\dots$	$e_{(\alpha-1)K_C+2\beta-1}$	0
e_{β}	$e_{\beta-1}$		0	$e_{2\beta}$	$e_{K_C+2\beta}$		$e_{(\alpha-1)K_C+2\beta}$	0
0	e_{β}		0	$e_{2\beta+1}$	$e_{K_C+2\beta+1}$		$e_{(\alpha-1)K_C+2\beta+1}$	0
$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
0	0		e_1	$e_{\beta+K_C}$	$e_{2K_C+\beta}$		$e_{\alpha K_C+\beta}$	0

TABLE VII
MATRIX E WHEN USER WANTS W_1 AND $N - K_C - T + 1 > K_C$ (CASE 2).

by $S_{n,r}$, the K_C random variables $\{S_{n,1}, \dots, S_{n,K_C}\} = \mathbf{S}_n$ are the randomness stored at node n . The answer sent back to the user by node n in section r is generated by

$$A_n^{[1],(r)} = \langle Q_n^{[1],(r)}, D_n \rangle + \underbrace{\phi_n \psi_n \sum_{j=1}^{K_C+T-1} S_j^{(r)} \cdot \lambda_n^{j-1}}_{S_{n,r}}.$$

Because we use an MDS code of dimension T in the generation of queries, every T nodes receive statistically uniformly random query vectors. Hence, user-privacy is guaranteed.

To show that the user can decode W_1 successfully and database-privacy is guaranteed, we illustrate via transmission in section 1 under Case 1. Let $E_n^{(1)}$ denote the vector of the first $(N - K_C - T + 1)K$ entries in E_n , the inner products $\langle E_n^{(1)}, D_n \rangle$ for all N nodes retrieve $N - K_C - T + 1$ linear combinations of different symbols from W_1 .

$$\begin{aligned} A_n^{[1],(1)} &= \langle Q_n^{[1],(1)}, D_n \rangle + \phi_n \psi_n \sum_{j=1}^{K_C+T-1} S_j^{(1)} \cdot \lambda_n^{j-1} \\ &= \langle \tilde{U}_n^{(1)} + E_n^{(1)}, D_n \rangle + \phi_n \psi_n \sum_{j=1}^{K_C+T-1} S_j^{(1)} \cdot \lambda_n^{j-1} \\ &= \phi_n \psi_n \sum_{l=1}^{(N-K_C-T+1)K} (\mathbf{U}_{l,1}^{(1)} \cdot 1 + \dots + \mathbf{U}_{l,T}^{(1)} \cdot \lambda_n^{T-1}) \\ &\quad \cdot (\mathbf{W}_{l,1} \cdot 1 + \dots + \mathbf{W}_{l,K_C} \cdot \lambda_n^{K_C-1}) + \langle E_n^{(1)}, D_n \rangle \\ &\quad + \phi_n \psi_n \sum_{j=1}^{K_C+T-1} S_j^{(1)} \cdot \lambda_n^{j-1} \\ &= \phi_n \psi_n \sum_{j=1}^{K_C+T-1} (X_j^{(1)} + S_j^{(1)}) \cdot \lambda_n^{j-1} + \langle E_n^{(1)}, D_n \rangle, \end{aligned}$$

where denoting the columns of $\mathbf{W}$ by $\mathbf{W}_1, \dots, \mathbf{W}_{K_C}$,

$$X_j^{(1)} = \sum_{\substack{t+m=j+1 \\ 1 \leq t \leq T, 1 \leq m \leq K_C}} \langle \mathbf{U}_t^{(1)}, \mathbf{W}_m \rangle$$

Hence, in section 1, the user downloads N symbols, which are independent linear combinations of $K_C + T - 1$ unknowns $(X_j^{(1)} + S_j^{(1)})$'s and $N - K_C - T + 1$ unknowns $\langle E_n^{(1)}, D_n \rangle$. Therefore, the user can solve the linear system and obtain $N - K_C - T + 1$ linear combinations of different symbols from W_1 . The database privacy is guaranteed because of the randomness $S_j^{(1)}$. Over all K_C sections, the user obtain $K_C(N - K_C - T + 1)$ independent linear combinations of symbols

of W_1 , which are sufficient to decode the desired file W_1 .

VIII. DISCUSSION: SPIR WITH MDS-CODED FILES, AND COMPLETE RANDOMNESS SHARED BY ALL SERVERS

In this section, we discuss the case where only the files are MDS-coded and distributed among the servers, but complete randomness information is shared by all servers.⁵ This is motivated by two practical scenarios: 1) the scenario where all randomness information is stored at each server *a priori*, before the phase when the files are coded and stored in the distributed storage system; 2) the scenario where the servers have access to the same seeds for generating the same randomness information to be used in the answers. This model is different from the assumption of the main part of this work (with all data MDS-coded, including both files and randomness) and is motivated by a different kind of practical system. We show our current capacity results, including an example with $N = 3, K_C = 2, K = 2$ in Section VIII-B and the special case with $K_C = N$ in Section VIII-C. Before going into the details of the examples, in Section VIII-A below, we summarize the differences in the capacity results between the model with shared randomness and the model with coded randomness. From the capacity results of the example and the special case, it can be observed that with the same randomness at all servers, higher capacity can be achieved. Surprisingly, the capacity of the special case depends on the number of messages K , which is different from all previous SPIR capacities [7], [19], and Theorems 1-3.⁶ The capacity for general parameters with shared randomness remains an open problem.

A. Observation and Discussion

Compared to the case with both files and randomness MDS-coded (Theorem 1), the case with MDS-coded files and complete randomness shared by all servers is fundamentally different, from not only the motivation and modeling perspective, but also the theoretical perspective and the resulting fundamental limits of the problems.

- 1) *Capacity depends on K , different from previous SPIR capacity results:* From the special case in Section VIII-C, the SPIR capacity depends on the number of messages K when files are coded but randomness is shared. This is different from previous SPIR capacity results [7], [19], and Theorems 1-3. We conjecture that the SPIR

⁵We consider non-colluding servers in this section.

⁶The same phenomenon starts to appear in more complex generalization in recent work [20] (Corollary 4).

capacity for general parameters with coded files and shared randomness also depend on K , which remains an open problem.

- 2) *Different proof techniques used:* The proof techniques used in both achievable schemes and converse proofs differ from the techniques used for the problem with coded randomness. In the optimal schemes with coded randomness, the user randomizes only the way file symbols are mixed in the answers, and the same fixed randomness is added to the answers independent of the queries. In the examples below, when the randomness is shared by all servers, the user can randomize the way that the randomness is added to the answers to achieve higher rates (through random permutations of the randomness). The converse proof techniques for coded randomness can no longer be used here, because the data available at each K_C nodes are no longer statistically independent.
- 3) *Higher achievable rate:* In Section VIII-B, we show that when $N = 3, K_C = 2, K = 2$, the capacity is $1/2$, which is higher than $C_{\text{MDS-SPiR}} = 1/3$. Similarly, when $K_C = N$, positive capacity is achievable from Section VIII-C, which is higher than the capacity $C_{\text{MDS-SPiR}} = 0$ for (N, K_C) -MDS coded randomness. In the achievable schemes presented above, the user randomizes the randomness symbol s added to the message symbols. The utility of this strategy is twofold: to guarantee user-privacy; and to achieve higher rate by aligning the interference of the “randomly added” randomness.
- 4) *Less total amount of randomness needed:* When $N = 3, K_C = 2, K = 2$, the relative amount of total randomness needed to achieve positive rate is $\rho = 1$, which is smaller than $\frac{K_C}{N-K_C} = 2$ for coded randomness. For the case when $K_C = N$, the least amount of total randomness needed is $\rho = \frac{H(S)}{L} \geq \frac{KN-K-N+2}{N-1}$, which is smaller than ∞ for coded randomness (i.e., no positive rate achievable).

B. Example: $N = 3$ servers, $K_C = 2$, $K = 2$ files

We show that the capacity of SPiR with coded files and shared randomness with $N = 3$, $K_C = 2$ and $K = 2$ is

$$C = \begin{cases} 1/2, & \text{if } \rho \geq 1 \\ 0, & \text{otherwise} \end{cases}.$$

1) *Achievable scheme:* Assume each file consists of 6 symbols, denoted by $W_1 = (a_1, a_2, a_3, a_4, a_5, a_6)$, $W_2 = (b_1, b_2, b_3, b_4, b_5, b_6)$. The files are distributedly stored at the 3 nodes by a $(3, 2)$ -MDS code, represented by

node 1	node 2	node 3
a_1	a_2	$a_1 + a_2$
a_3	a_4	$a_3 + a_4$
a_5	a_6	$a_5 + a_6$
b_1	b_2	$b_1 + b_2$
b_3	b_4	$b_3 + b_4$
b_5	b_6	$b_5 + b_6$

All 3 nodes share the same set of uniform i.i.d. random

symbols $\{s_1, s_2, \dots, s_6\}$. W.l.o.g, suppose W_1 is desired. Let $\pi : [1 : 6] \rightarrow [1 : 6]$ be randomly drawn from all permutations of $[1 : 6]$, the retrieval scheme is described by the table below.

node 1	node 2	node 3
$a_1 + s_{\pi(1)}$	$a_4 + s_{\pi(1)}$	$s_{\pi(1)}$
$b_1 + s_{\pi(2)}$	$b_2 + s_{\pi(3)}$	$a_3 + a_4 + b_1 + b_2 + s_{\pi(2)} + s_{\pi(3)}$
$b_3 + s_{\pi(4)}$	$b_4 + s_{\pi(5)}$	$a_1 + a_2 + b_3 + b_4 + s_{\pi(4)} + s_{\pi(5)}$
$a_5 + s_{\pi(6)}$	$a_6 + s_{\pi(6)}$	$s_{\pi(6)}$

User-privacy holds because each node observes the same number of independent symbols from each file, and randomly independent s symbols. It can be checked that the user obtains no information regarding $b_{[1:6]}$, hence database-privacy is guaranteed. The scheme achieves the rate $R = 6/12 = 1/2$, with $\rho = 6/6 = 1$.

2) *Converse:*

$$\begin{aligned} L &= H(W_1) \\ &= H(W_1) - H(W_1 | A_{[1:3]}^{[1]}, Q_{[1:3]}^{[1]}) \\ &= H(A_{[1:3]}^{[1]} | Q_{[1:3]}^{[1]}) - H(A_{[1:3]}^{[1]} | Q_{[1:3]}^{[1]}, W_1) \\ &= H(A_{[1:3]}^{[1]} | Q_{[1:3]}^{[1]}) - H(A_1^{[1]} | Q_{[1:3]}^{[1]}, W_1) \\ &\quad - H(A_2^{[1]} | A_1^{[1]}, Q_{[1:3]}^{[1]}, W_1) - H(A_3^{[1]} | A_1^{[1]}, A_2^{[1]}, Q_{[1:3]}^{[1]}, W_1) \\ &\leq H(A_{[1:3]}^{[1]} | Q_{[1:3]}^{[1]}) - H(A_1^{[1]} | Q_{[1:3]}^{[1]}, W_1) \\ &\quad - H(A_2^{[1]} | A_1^{[1]}, Q_{[1:3]}^{[1]}, W_1) \\ &= H(A_{[1:3]}^{[1]} | Q_{[1:3]}^{[1]}) - H(A_1^{[1]} | Q_{[1:3]}^{[1]}) \\ &\quad - H(A_2^{[1]} | A_1^{[1]}, Q_{[1:3]}^{[1]}, W_1), \end{aligned}$$

Denote the storage of W_2 at node 1 as $W_{2,1}$ and that at node 2 as $W_{2,2}$,

$$\begin{aligned} &I(A_1^{[1]}; A_2^{[1]} | W_1, S, Q_{[1:3]}^{[1]}) \\ &\leq I(A_1^{[1]}, W_{2,1}; A_2^{[1]}, W_{2,2} | W_1, S, Q_{[1:3]}^{[1]}) \\ &= I(W_{2,1}; W_{2,2} | W_1, S, Q_{[1:3]}^{[1]}) \\ &= 0. \end{aligned}$$

Therefore,

$$\begin{aligned} H(A_2^{[1]} | A_1^{[1]}, Q_{[1:3]}^{[1]}, W_1) &\geq H(A_2^{[1]} | A_1^{[1]}, Q_{[1:3]}^{[1]}, W_1, S) \\ &= H(A_2^{[1]} | Q_{[1:3]}^{[1]}, W_1, S) \\ &= H(A_2^{[1]} | Q_1^{[1]}, W_1, S) \\ &= H(A_2^{[2]} | Q_1^{[2]}, W_1, S) \\ &= H(A_2^{[2]} | Q_{[1:3]}^{[2]}, W_1, S) \\ &\geq 1/3 H(A_{[1:3]}^{[2]} | Q_{[1:3]}^{[2]}, W_1, S) \\ &= 1/3 H(A_{[1:3]}^{[2]}, W_2 | Q_{[1:3]}^{[2]}, W_1, S) \\ &= 1/3 H(W_2) = 1/3 L. \end{aligned}$$

Therefore, $L \leq H(A_{[1:3]}^{[1]} | Q_{[1:3]}^{[1]}) - H(A_1^{[1]} | Q_{[1:3]}^{[1]}) - \frac{1}{3}L$. By replacing $H(A_1^{[1]} | Q_{[1:3]}^{[1]})$ with $H(A_1^{[n]} | Q_{[1:3]}^{[1]})$ for $n = 2, 3$,

averaging over n and applying Han's inequality,

$$L \leq (1 - 1/3)H(A_{[1:3]}^{[1]}|Q_{[1:3]}^{[1]}) - \frac{1}{3}L.$$

$$\text{Hence, } R = \frac{L}{H(A_1^{[1]})+H(A_2^{[1]})+H(A_3^{[1]})} \leq \frac{L}{H(A_{[1:3]}^{[1]}|Q_{[1:3]}^{[1]})} \leq 1/2.$$

By database-privacy,

$$\begin{aligned} 0 &= I(W_2, W_3; A_{[1:3]}^{[1]}|Q_{[1:3]}^{[1]}) \\ &= I(W_2, W_3; A_{[1:3]}^{[1]}|W_1, Q_{[1:3]}^{[1]}) \\ &= H(A_{[1:3]}^{[1]}|W_1, Q_{[1:3]}^{[1]}) - H(A_{[1:3]}^{[1]}|W_1, W_2, W_3, Q_{[1:3]}^{[1]}) \\ &\quad + H(A_{[1:3]}^{[1]}|W_1, W_2, W_3, S, Q_{[1:3]}^{[1]}) \\ &= H(A_{[1:3]}^{[1]}|W_1, Q_{[1:3]}^{[1]}) - I(S; A_{[1:3]}^{[1]}|W_1, W_2, W_3, Q_{[1:3]}^{[1]}) \\ &\geq H(A_{[1:3]}^{[1]}|W_1, Q_{[1:3]}^{[1]}) - H(S). \end{aligned}$$

$$\text{Therefore, } H(S) \geq H(A_{[1:3]}^{[1]}|W_1, Q_{[1:3]}^{[1]}) = H(A_{[1:3]}^{[1]}|Q_{[1:3]}^{[1]}) - L \geq L, \text{ and } \rho = \frac{H(S)}{L} = 1.$$

C. Special case when $K_C = N$, arbitrary number of files K

We show that the capacity of SPIR with coded files and shared randomness with $K_C = N$ and arbitrary K is

$$C = \begin{cases} \frac{N-1}{KN-K+1}, & \text{if } \rho \geq \frac{KN-K-N+2}{N-1} \\ 0, & \text{otherwise} \end{cases}.$$

1) *Achievable scheme:* We start with an example with $N = K_C = 2$, $K = 2$ files. Let $W_1 = (a_1, a_2)$, $W_2 = (b_1, b_2)$. The 2 nodes share 4 i.i.d. random variables s_1, s_2, s_3, s_4 . Let $\pi : [1 : 4] \rightarrow [1 : 4]$ be randomly drawn from all permutations of $[1 : 4]$.

Suppose W_1 is wanted, the retrieval scheme is described in the table below.

node 1	node 2
$a_1 + s_{\pi(1)}$	$s_{\pi(1)}$
$b_1 + s_{\pi(2)}$	$a_2 + s_{\pi(3)}$
$s_{\pi(3)}$	$b_2 + s_{\pi(4)}$

User-privacy is guaranteed because each node observes one symbol from each message, and independent s symbols. Because $s_{\pi(2)}, s_{\pi(4)}$ are independent, the user obtains no information regarding (b_1, b_2) . Therefore, database-privacy is guaranteed. The scheme achieves the rate $R = 2/6 = 1/3$.

From this example, we see that when $K_C = N$, positive rate can still be achieved with shared randomness. This is different from the case when both files and randomness are MDS-coded (Theorem 1), where the capacity is 0 when $K_C = N$. Next, we generalize the scheme for arbitrary parameters $K_C = N$ and K .

Assume each file consists of $L = N(N-1)$ symbols, and is denoted by

$$W_k = \begin{bmatrix} w_{k,1}^1 & w_{k,2}^1 & \cdots & w_{k,N}^1 \\ w_{k,1}^2 & w_{k,2}^2 & \cdots & w_{k,N}^2 \\ \vdots & \vdots & \ddots & \vdots \\ w_{k,1}^{N-1} & w_{k,2}^{N-1} & \cdots & w_{k,N}^{N-1} \end{bmatrix},$$

where the n th column denotes the symbols of W_k stored at node n . All N nodes share $N(KN-K-N+2)$ uniformly random and independent symbols $S_1, S_2, \dots, S_{N(KN-K-N+2)}$. W.l.o.g., assume W_1 is the desired file. Let $\pi : [1 : N(KN-K-N+2)] \rightarrow [1 : N(KN-K-N+2)]$ be a permutation mapping drawn uniformly at random from all permutations of $[1 : N(KN-K-N+2)]$. The retrieval scheme is represented in Table VIII.

It is evident that W_1 can be retrieved, because the N random symbols $S_{\pi(1)}, \dots, S_{\pi(N)}$ which are added to the symbols of W_1 are downloaded directly. User-privacy is guaranteed because each individual node observes linearly independent S symbols. Database-privacy is guaranteed because the symbols of undesired files are all added with independent S symbols.

The scheme achieves a rate $R = \frac{N(N-1)}{N(N+(K-1)(N-1))} = \frac{N-1}{KN-K+1}$, and $\rho = \frac{N(KN-K-N+2)}{N(N-1)} = \frac{KN-K-N+2}{N-1}$.

2) *Converse:* W.l.o.g., assume W_1 is desired.

$$\begin{aligned} L &= H(W_1) \\ &\stackrel{(a)}{=} H(W_1|Q) - H(W_1|A_{[1:N]}^{[1]}, Q) \\ &= H(A_{[1:N]}^{[1]}|Q) - H(A_{[1:N]}^{[1]}|W_1, Q) \\ &= H(A_{[1:N]}^{[1]}|Q) - H(A_1^{[1]}|W_1, Q) - H(A_{[2:N]}^{[1]}|A_1^{[1]}, W_1, Q) \\ &\stackrel{(b)}{=} H(A_{[1:N]}^{[1]}|Q) - H(A_1^{[1]}|Q) - H(A_{[2:N]}^{[1]}|A_1^{[1]}, W_1, Q), \end{aligned} \tag{24}$$

where step (a) follows from 3. Step (b) follows from user-privacy and database-privacy, such that $H(A_1^{[1]}|W_1, Q) = H(A_1^{[2]}|W_1, Q) = H(A_1^{[2]}|Q) = H(A_1^{[1]}|Q)$.

Denote the data stored at node n on W_k by $W_{k,n}$, by database-privacy,

$$\begin{aligned} &H(A_{[2:N]}^{[1]}|A_1^{[1]}, W_1, Q) \\ &= H(A_{[2:N]}^{[1]}|A_1^{[1]}, W_1, W_{[2:K],1}, Q) \\ &\geq H(A_{[2:N]}^{[1]}|A_1^{[1]}, W_1, W_{[2:K],1}, S, Q) \\ &\stackrel{(a)}{=} H(A_{[2:N]}^{[1]}|W_1, W_{[2:K],1}, S, Q) \\ &\stackrel{(b)}{=} \sum_{n=2}^N H(A_n^{[1]}|W_1, W_{[2:K],1}, S, Q) \\ &\stackrel{(c)}{=} \sum_{n=2}^N H(A_n^{[k]}|W_1, W_{[2:K],1}, S, Q) \\ &\stackrel{(d)}{=} H(A_{[2:N]}^{[k]}|W_1, W_{[2:K],1}, S, Q) \\ &\stackrel{(e)}{=} H(A_{[2:N]}^{[k]}|W_1, W_{[2:K],1}, S, Q) - H(A_{[2:N]}^{[k]}|W_{[1:K]}, S, Q) \\ &= H(W_{[2:K], [2:N]}|W_1, W_{[2:K],1}, S, Q) \\ &\quad - H(W_{[2:K], [2:N]}|A_{[2:N]}^{[k]}, W_1, W_{[2:K],1}, S, Q) \\ &\stackrel{(f)}{=} H(W_{[2:K], [2:N]}|W_1, W_{[2:K],1}, S, Q) \\ &= \frac{(K-1)(N-1)}{N}L, \end{aligned}$$

where step (a) holds because $A_1^{[1]}$ is a deterministic function of the queries, S , and the data stored at node 1 $W_{[1:K],1}$. Step (b) holds because given the queries and S , the answers of nodes $[2 : N]$ depend only on the data

node 1	node 2	...	node N
$S_{\pi(N)}$	$S_{\pi(1)}$	...	$S_{\pi(N-1)}$
$w_{1,1}^1 + S_{\pi(1)}$	$w_{1,2}^1 + S_{\pi(2)}$	...	$w_{1,N}^1 + S_{\pi(N)}$
$w_{1,1}^2 + S_{\pi(2)}$	$w_{1,2}^2 + S_{\pi(3)}$	...	$w_{1,N}^2 + S_{\pi(1)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$w_{1,1}^{N-1} + S_{\pi(N-1)}$	$w_{1,2}^{N-1} + S_{\pi(N)}$	...	$w_{1,N}^{N-1} + S_{\pi(N-2)}$
$w_{2,1}^1 + S_{\pi(N+1)}$	$w_{2,2}^1 + S_{\pi(N+2)}$	...	$w_{2,N}^1 + S_{\pi(2N)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$w_{2,1}^{N-1} + S_{\pi(N^2-N+1)}$	$w_{2,2}^{N-1} + S_{\pi(N^2-N+2)}$	...	$w_{2,N}^{N-1} + S_{\pi(N^2)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$w_{K,1}^1 + S_{\pi((K-2)N^2-(K-3)N+1)}$	$w_{K,2}^1 + S_{\pi((K-2)N^2-(K-3)N+2)}$	...	$w_{K,N}^1 + S_{\pi((K-2)N^2-(K-4)N)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$w_{K,1}^{N-1} + S_{\pi(KN^2-N^2-KN+N+1)}$	$w_{K,2}^{N-1} + S_{\pi(KN^2-N^2-KN+N+2)}$	...	$w_{K,N}^{N-1} + S_{\pi(KN^2-N^2-KN+2N)}$

TABLE VIII
RETRIEVAL SCHEME FOR SHARED RANDOMNESS AND $K_C = N$.

they store, which are statistically independent. Step (c) follows from user privacy, in which the file index k can be arbitrary from $[1 : K]$. Step (d) follows for similar reason as (b). Step (e) holds because the answers are deterministic functions of all files $W_{[1:K]}$, S , and the queries. Step (f) holds because $A_1^{[1]}$ is a deterministic function of $W_1, W_{[2:K],1}, S, Q$, and k can be arbitrary from $[1 : K]$, therefore, $H(W_{[2:K],[2:N]}|A_{[2:N]}^{[k]}, W_1, W_{[2:K],1}, S, Q) = H(W_{[2:K],[2:N]}|A_{[1:N]}^{[k]}, W_1, W_{[2:K],1}, S, Q) = \sum_{k=2}^K H(W_{k,[2:N]}|A_{[1:N]}^{[k]}, W_{[1:k-1]}, W_{[k:K],1}, S, Q) = 0$ by decodability. Therefore, $L \leq H(A_{[1:N]}^{[1]}|Q) - H(A_1^{[1]}|Q) - \frac{(K-1)(N-1)}{N}L$, where $H(A_1^{[1]}|Q)$ can be replaced by $H(A_n^{[1]}|Q)$ for any $n \in [1 : N]$. Averaging over all $n \in [1 : N]$ and by Han's inequality,

$$L \leq (1 - \frac{1}{N})H(A_{[1:N]}^{[1]}|Q) - \frac{(K-1)(N-1)}{N}L. \quad (25)$$

$$\text{Hence, } R = \frac{L}{\sum_{n=1}^N H(A_n^{[k]}|Q)} \leq \frac{L}{H(A_{[1:N]}^{[1]}|Q)} \leq \frac{N-1}{KN-K+1}.$$

To obtain a lower bound on the amount of randomness, by database-privacy,

$$\begin{aligned} 0 &= I(W_{[2:K]}; A_{[1:N]}^{[1]}|Q) \\ &= H(W_{[2:K]}|Q) - H(W_{[2:K]}|A_{[1:N]}^{[1]}, W_1, Q) \\ &= H(A_{[1:N]}^{[1]}|W_1, Q) - H(A_{[1:N]}^{[1]}|W_{[1:K]}, Q) \\ &\quad + H(A_{[1:N]}^{[1]}|S, W_{[1:K]}, Q) \\ &= H(A_{[1:N]}^{[1]}|W_1, Q) - I(S; A_{[1:N]}^{[1]}, W_{[1:K]}, Q) \\ &\geq H(A_{[1:N]}^{[1]}|W_1, Q) - H(S). \end{aligned}$$

Combine with (24) and (25),

$$\begin{aligned} H(S) &\geq H(A_{[1:N]}^{[1]}|W_1, Q) = H(A_{[1:N]}^{[1]}|Q) - L \\ &\geq \frac{KN - K - N + 2}{N - 1}L. \end{aligned}$$

$$\text{Therefore, } \rho = \frac{H(S)}{L} \geq \frac{KN-K-N+2}{N-1}.$$

IX. CONCLUSION

In this paper, we study three variants of the SPIR problem, namely, SPIR with MDS-coded storage and non-colluding servers (MDS-SPIR), SPIR with replicated storage and colluding servers (TSPIR), and SPIR with MDS-coded storage and colluding servers (MDS-TSPIR). We derive the capacity of MDS-SPIR and TSPIR. For MDS-TSPIR, we restrict to schemes with additive randomness, where the servers add the randomness to the answers in a deterministic way, regardless of what queries are received. For schemes with more general randomness strategies, MDS-TSPIR capacity remains an open problem.⁷

As in the previous results on the classical PIR and SPIR problems [5], [7], SPIR capacity does not depend on the number of files K , and the PIR capacity approaches the SPIR capacity as the number of files tends to infinity. The same phenomenon can be observed with the variants considered in this work. The MDS-SPIR capacity is the limit of the MDS-PIR capacity [8] taking the number of files to infinity; the TSPIR is the limit of the TPIR capacity [6] for infinitely many files. The intuition of this observation in all the PIR and SPIR variants is that, when the number of files increases, the penalty in the downloading rate to protect database-privacy decays. When there are asymptotically infinitely many files, the information rate that the user can learn about the database from finite downloaded symbols vanishes. However, the phenomenon no longer holds when the randomness is shared by all servers, since then the SPIR capacity does depend on K , as can be concluded from the special cases studied in Section VIII-C.

⁷The PIR version, namely, the MDS-TPIR capacity for general parameters also remains an open problem. In [16], MDS-TPIR capacity for some special cases of the parameters, i.e. $(K, N, K_C, T) = (2, 4, 2, 2)$ and $(K, N, K_C, T) = (2, N, N-1, T)$, are derived.

For SPIR problems, two sources of information need to be stored/accessed among the servers, *i.e.*, the database (files) and the randomness. Different ways of storing the files and the randomness might be motivated from different kinds of practical systems. In previous SPIR works and this work, the storage for the files and the randomness are consistent. Specifically, when the files are replicated at all servers, it is assumed that the randomness is also replicatively stored at all servers [7], [19]; when the files are MDS-coded among the servers, any set of servers which store independent file information should also store independent randomness.⁸ When the storage of randomness is inconsistent with the storage of files, *e.g.*, the files are MDS-coded but the whole set of randomness is stored at each server, we found via examples that the problem becomes fundamentally different. The capacity for the general case with shared randomness is still an open problem. The capacity for the most general case with inconsistent storage is an interesting direction for future work.

ACKNOWLEDGMENT

The authors would like to thank Dr. Hua Sun from University of North Texas for the discussion on the storage of the randomness and its importance in SPIR problems.

REFERENCES

- [1] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," in *IEEE Annual Symposium on Foundations of Computer Science*, 1995, pp. 41–50.
- [2] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," *Journal of the ACM (JACM)*, 1998.
- [3] Y. Gertner, Y. Ishai, E. Kushilevitz, and T. Malkin, "Protecting data privacy in private information retrieval schemes," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, 1998.
- [4] W. Gasarch, "A survey on private information retrieval," in *Bulletin of the EATCS*. Citeseer, 2004.
- [5] H. Sun and S. A. Jafar, "The capacity of private information retrieval," *IEEE Transactions on Information Theory*, vol. 63, no. 7, pp. 4075–4088, 2017.
- [6] —, "The capacity of robust private information retrieval with colluding databases," *IEEE Transactions on Information Theory*, vol. 64, no. 4, pp. 2361–2370, 2018.
- [7] —, "The capacity of symmetric private information retrieval," *IEEE Transactions on Information Theory*, 2018.
- [8] K. Banawan and S. Ulukus, "The capacity of private information retrieval from coded databases," *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1945–1956, 2018.
- [9] —, "Multi-message private information retrieval: Capacity results and near-optimal schemes," *IEEE Transactions on Information Theory*, 2018.
- [10] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4539–4551, 2010.
- [11] N. B. Shah, K. Rashmi, and K. Ramchandran, "One extra bit of download ensures perfectly private information retrieval," in *Proc. IEEE Int. Symp. Information Theory*, 2014, pp. 856–860.
- [12] A. Fazeli, A. Vardy, and E. Yaakobi, "PIR with low storage overhead: coding instead of replication," *arXiv preprint arXiv:1505.06241*, 2015.
- [13] T. H. Chan, S.-W. Ho, and H. Yamamoto, "Private information retrieval for coded storage," in *Proc. IEEE Int. Symp. Information Theory*, 2015, pp. 2842–2846.
- [14] R. Tajeddine and S. E. Rouayheb, "Private information retrieval from MDS coded data in distributed storage systems," in *Proc. IEEE Int. Symp. Information Theory*, 2016.
- [15] R. Freij-Hollanti, O. W. Gnilke, C. Hollanti, and D. A. Karpuk, "Private information retrieval from coded databases with colluding servers," *SIAM Journal on Applied Algebra and Geometry*, vol. 1, no. 1, pp. 647–664, 2017.
- [16] H. Sun and S. A. Jafar, "Private information retrieval from MDS coded data with colluding servers: Settling a conjecture by Freij-Hollanti et al.," *IEEE Transactions on Information Theory*, vol. 64, no. 2, pp. 1000–1022, 2018.
- [17] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, 2012.
- [18] D. Mirandola and G. Zémor, "Critical pairs for the product singleton bound," *IEEE Transactions on Information Theory*, vol. 61, no. 9, pp. 4928–4937, 2015.
- [19] Q. Wang and M. Skoglund, "Secure symmetric private information retrieval from colluding databases with adversaries," in *55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2017, pp. 1083–1090.
- [20] Z. Jia, H. Sun, and S. A. Jafar, "Cross subspace alignment and the asymptotic capacity of x -secure t -private information retrieval," *arXiv preprint arXiv:1808.07457*, 2018.
- [21] Q. Wang and M. Skoglund, "Symmetric private information retrieval for MDS coded distributed storage," in *IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

Qiwen Wang (M'18) received her B.Sc. degree in Mathematics and B.Eng. degree in Information Engineering in 2010, and Ph.D. degree in Information Engineering in 2015, all from the Chinese University of Hong Kong, Hong Kong. She is currently a postdoctoral researcher in the Department of Information Science and Engineering, KTH Royal Institute of Technology, Stockholm, Sweden.

Mikael Skoglund (S'93-M'97-SM'04-F'19) received the Ph.D. degree in 1997 from Chalmers University of Technology, Sweden. In 1997, he joined the Royal Institute of Technology (KTH), Stockholm, Sweden, where he was appointed to the Chair in Communication Theory in 2003. At KTH, he heads the Department of Information Science and Engineering.

Dr. Skoglund has worked on problems in source-channel coding, coding and transmission for wireless communications, Shannon theory, information and control, and statistical signal processing. He has authored and co-authored some 150 journal and 350 conference papers.

Dr. Skoglund has served on numerous technical program committees for IEEE sponsored conferences. During 2003–08 he was an associate editor with the IEEE Transactions on Communications and during 2008–12 he was on the editorial board for the IEEE Transactions on Information Theory. He is general co-chair for IEEE ITW 2019.

⁸We would like to clarify that in our previous conference paper [21], the randomness is shared by all servers, but we require that the user shall not learn any information about the randomness. In this paper, we remove the privacy requirement on the user not learning the randomness, but the randomness should also be stored in a distributed way, which in our opinion is more realistic for distributed storage systems. The capacity result remains the same.