Double Blind T-Private Information Retrieval

Yuxiang Lu, *Graduate Student Member, IEEE*, Zhuqing Jia, *Graduate Student Member, IEEE*, and Syed A. Jafar, *Fellow, IEEE*

***Abstract*—Double blind T-private information retrieval (DB-TPIR) enables two users, each of whom specifies an index (θ_1, θ_2, resp.), to efficiently retrieve a message $W(\theta_1, \theta_2)$ labeled by the two indices, from a set of N servers that store all messages $W(k_1, k_2), k_1 \in \{1, 2, \ldots, K_1\}, k_2 \in \{1, 2, \ldots, K_2\}$, such that the two users' indices are kept private from any set of up to T_1, T_2 colluding servers, respectively, as well as from each other. A DB-TPIR scheme based on cross-subspace alignment is proposed in this paper, and shown to be capacity-achieving in the asymptotic setting of large number of messages and bounded latency. The scheme is then extended to M-way blind X-secure T-private information retrieval (MB-XS-TPIR) with multiple (M) indices, each belonging to a different user, arbitrary privacy levels for each index ($T_1, T_2, \ldots, T_M$), and arbitrary level of security (X) of data storage, so that the message $W(\theta_1, \theta_2, \ldots, \theta_M)$ can be efficiently retrieved while the stored data is held secure against collusion among up to X colluding servers, the m^{th} user's index is private against collusion among up to T_m servers, and each user's index θ_m is private from all other users. The general scheme relies on a tensor-product based extension of cross-subspace alignment and retrieves $1 - (X + T_1 + \cdots + T_M)/N$ bits of desired message per bit of download.**

***Index Terms*—Secure storage, privacy, capacity.**

I. Introduction

DATA privacy and security are among the biggest challenges of the modern information age. Driven by these challenges there is much interest in the building blocks (primitives) of privacy/security preserving schemes, such as secret sharing [1], oblivious transfer [2], private information retrieval (PIR) [3], [4], secure multiparty computation (MPC) [5]–[7], and private simultaneous messages (PSM) [8]. Understanding the fundamental limits of each of these building blocks is the key to understanding the scope of their potential applications. The focus of this work is on private information retrieval (PIR).

Introduced by Chor *et al.* in [3], [4], the goal of PIR in its simplest form is to allow a user to efficiently retrieve a desired message from a set of K messages that are replicated across N distributed servers, while revealing no information to any individual server about which message is desired. Until recently, PIR was investigated primarily by computer scientists and cryptographers [3], [4] under the assumption of *short* messages (e.g., each message is just one bit), with the goal of minimizing the total communication (upload and download) cost. However, following the capacity characterization of PIR in [9], [10] under the assumption of *long* messages (where downloads dominate the communication cost), the fundamental limits (capacity) of various forms of download-efficient PIR have become an active topic in information theory. Recent advances include the capacity characterizations of PIR with T-privacy [11], symmetric-privacy [12], weak privacy [13], [14], eavesdroppers and/or Byzantine servers [15]–[19], coded storage [20]–[27], secure storage [28]–[30], limited storage [31]–[35], cached data or side information [36]–[39], multiple rounds [40], [41], multiple desired messages [42]–[45], upload constraints [46], arbitrary collusion patterns [21], [47], single server PIR with user side information [48]–[54], latent-variable single server PIR [55], as well as applications of PIR to private computation [56]–[59], private search [60], private set intersection [45], coded computing [61], locally decodable codes [62], etc.

Our goal in this work is to further expand the understanding of download-efficient PIR in a new direction—*M-way blind X-secure T-PIR* or MB-XS-TPIR, where the data, labeled by M indices, is stored in an X-secure[1] fashion by N servers, and M users jointly retrieve a desired message by specifying one index each (user m specifies θ_m $\forall m \in \{1, 2, \ldots, M\}$), while keeping their index private from each other and also T-private from the servers where the data is stored. It is conceivable that such a functionality may be directly useful. For example, consider private data, e.g., health records, that are stored anonymously and X-securely among a cloud of distributed servers. For enhanced security it is not uncommon to require multi-factor authentication, e.g., 2-factor authentication from a pair of devices (say, smartphone and computer) that belong to the owner of the data (patient) in order to allow access to the data. This can be implemented as the double blind setting of MB-XS-TPIR by creating 2 passwords (indices θ_1, θ_2), so that the two devices must each provide θ_1, θ_2 respectively, in order for the patient to retrieve $\mathbf{W}(\theta_1, \theta_2)$ on either device. It is important that each device learns nothing about the other device's password (treating devices as users, this is called inter-user privacy), so that the loss or hacking of either device

Manuscript received August 13, 2020; revised November 30, 2020 and January 18, 2021; accepted January 18, 2021. Date of publication January 22, 2021; date of current version March 16, 2021. The work of Yuxiang Lu, Zhuqing Jia, and Syed A. Jafar was supported in part by the National Science Foundation under Grant CCF-1617504, Grant CCF-1907053, Grant CNS-1731384; in part by the Office of Naval Research under Grant N00014-18-1-2057; and in part by the Army Research Office under Grant W911NF-17-S-0002-03. *(Corresponding author: Yuxiang Lu.)*

The authors are with the Center for Pervasive Communications and Computing, Department of Electrical Engineering and Computer Science, University of California at Irvine, Irvine, CA 92697 USA (e-mail: yuxiang.lu@uci.edu; zhuqingj@uci.edu; syed@uci.edu).

Digital Object Identifier 10.1109/JSAIT.2021.3053481

[1] X-security (T-privacy) means that security (privacy) is guaranteed against any set of up to X (T) colluding servers.

does not reveal more than its own password. Furthermore, the passwords/indices are also kept T-private from the servers, so that even the servers learn nothing about which record is being retrieved. M-way authentication similarly motivates MB-XS-TPIR. In general, MB-XS-TPIR may be a good solution for secret sharing among multiple parties when the size of the secret is too large so that it needs to be securely stored among distributed servers (cloud) while access to the secret is allowed by distributing smaller keys or passwords (indices in MB-XS-TPIR) to the parties. The multiway blind functionality is also useful for secure multiparty computation[2] where the inputs $\theta_1, \ldots, \theta_M$ of a function $f(x_1, \ldots, x_M)$ are held by M parties and $\mathbf{W}$, whose $(\theta_1, \ldots, \theta_M)^{th}$ entry is the evaluation of the function at $(\theta_1, \ldots, \theta_M)$, is stored by distributed servers [63]. Fundamentally, however, our motivation is simply to expand the scope of a basic primitive.

The main contribution of this work is a cross-subspace alignment (CSA) based scheme for MB-XS-TPIR. To place this in perspective, we note that the evolution of CSA codes has followed a remarkable trajectory with crossovers between PIR and coded distributed computing (CDC). In a nutshell, CSA codes originated in PIR, then crossed over to CDC where the constructions were generalized, and now in this work, return back to PIR in their generalized form which allows MB-XS-TPIR. To see this in a bit more detail, recall that the idea of cross-subspace alignment originated in the context of XS-TPIR [29], [30] as a way to align interference from undesired product terms that result when a secret-shared (private) query vector is multiplied with a secret-shared (secure) data vector. It was then observed in [29], [44], [61], [64], [65] that the idea of aligning undesired product terms is similarly useful in distributed computing applications, which led to a crossover of CSA codes to coded distributed computing [66]. Generalized CSA codes were constructed in [61] to unify and improve upon several state-of-art CDC approaches like Lagrange Coded Computing [67] and Entangled Polynomial codes [68]. The generalized forms of CSA codes allow not only pairwise matrix multiplications, but also multilinear computations. This work represents the next step forward, as the generalizations of CSA codes that emerged in the context of coded distributed computing are used to enable new forms of PIR. Indeed, the main idea behind this work is the framing of a particular solution[3] to MB-XS-TPIR as a problem of distributed secure tensor product computation. With this mapping we find that the key to the solution is to compute the tensor products of suitably structured secret-shared query vectors that originate at the users, and correspondingly structured secret-shared data matrices that are stored at the servers. Note that CSA codes allow a range of structures corresponding to various choices of feasible code parameters, which may be further optimized for download cost depending on the application. See Section V-B for additional details. The desired tensor-products turn out to be multilinear operations, so that the multilinear computation capability of CSA codes can be applied to MB-XS-TPIR.

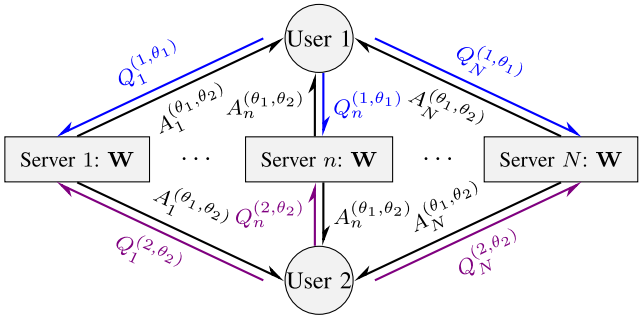

Fig. 1. The double blind T-private information retrieval (DB-TPIR) problem.

In order to introduce our solution in a more transparent setting, our initial focus is on DB-TPIR, i.e., the double-blind setting ($M = 2$) with T-private user indices (T_1, T_2, resp.) and replicated data storage, initially with no data-security, i.e., $X = 0$. This basic setting allows us to convey the main ideas behind the construction of the scheme and also to explore its optimality. Specifically, for the DB-TPIR problem we propose a scheme based on cross-subspace alignment [61] which allows the retrieval of $1-(T_1+T_2)/N$ bits of desired message per bit of download, regardless of the number of messages. By noting connections between this problem and X-secure T-private information retrieval (XS-TPIR) [29] we show that $1-(T_1+T_2)/N$ is also the asymptotic capacity of DB-TPIR as the number of messages approaches infinity, provided that the number of bits of each message that are jointly encoded is bounded (say, due to latency constraints).

With the insights obtained from DB-TPIR, we are then able to fully generalize our achievable scheme to MB-XS-TPIR, i.e., M-way blind X-secure T-private information retrieval with multiple (M) indices, each specified privately by a different user, arbitrary privacy levels for each index $(T_1, T_2, \ldots, T_M)$, and arbitrary level of security (X) of data storage, so that the message $W(\theta_1, \theta_2, \ldots, \theta_M)$ can be efficiently retrieved by the users while the stored data is held secure against collusion among up to X colluding servers, the m^{th} user's index is private against collusion among up to T_m servers, and each user's index θ_m is private from all other users. The general setting is based on an M-way tensor-product extension of cross-subspace alignment codes, and retrieves $1-(X+T_1+\cdots+T_M)/N$ bits of desired message per bit of download. This generalizes the known asymptotically (large number of messages) optimal schemes for various special cases of MB-XS-TPIR including DB-TPIR ($M = 2, X = 0$) and XS-TPIR ($M = 1$) [29] (which automatically recovers asymptotically optimal schemes for TPIR ($X = 0, M = 1$) [11] and PIR ($X = 0, M = 1, T_1 = 1$) [10] as well). In fact, the achievable scheme for MB-XS-TPIR also satisfies symmetric-privacy, i.e., the users learn nothing about the database or each others' indices, beyond the desired message. Therefore, it also

[2]A notable limitation is that M-way blind PIR allows communication only between users and servers, but Secure MPC protocols may in general also allow direct communication between users.

[3]The problem of MB-XS-TPIR, or PIR in general, is not *equivalent* to distributed matrix (tensor) multiplication. For example, there is no constraint in PIR that forces the answers returned by the servers to be linear in either the query vectors or the stored information, or more specifically, products of query vectors and the stored information. However, many *solutions* to PIR indeed take this form, thus creating a connection between PIR and CDC. That such solutions tend to be optimal in many cases strengthens this connection.

yields symmetrically private schemes as special cases. For example, the general MB-XS-TPIR scheme yields a capacity achieving scheme for Symmetric XS-TPIR ($M = 1$) [65], STPIR ($M = 1, X = 0$, Symmetric Privacy) [22] and SPIR ($M = 1, X = 0, T_1 = 1$) as well. Based on all these observations, we conjecture that the general MB-XS-TPIR scheme is also asymptotically optimal.

In order to compare the new scheme with state of art, a natural baseline is obtained from [63] where a secure multiparty computation (MPC) scheme is constructed based on symmetric-PIR (SPIR) as a building block. This construction can be naturally generalized to a DBPIR scheme. Intuitively, this construction is based on a partitioning of N servers into $\sqrt{N}$ groups of $\sqrt{N}$ servers each, such that within each sub-group the SPIR scheme is executed for one user, while across sub-groups the SPIR scheme is executed for the other user. However, even with the most efficient SPIR scheme as the building block, the rate of this construction for DBPIR is $(1-1/\sqrt{N})^2$, which is strictly smaller than the rate $1-2/N$ achieved by our asymptotically optimal scheme. This is because cross-subspace alignment allows us to avoid the 2-way partitioning of servers and is able to gain significant efficiency by jointly exploiting all servers. For example, with $N = 4$ servers, the partitioning based approach achieves a rate of $(1 - 1/\sqrt{N})^2 = 1/4$, while the new scheme achieves a 100% higher rate of $1 - 2/N = 1/2$ due to cross-subspace alignment.

This paper is organized as follows. Section II formalizes the general MB-XS-TPIR problem. Section III states the main results of this paper in the form of two theorems. Their proofs are presented in Section IV and Section V. Section VI concludes the paper.

Notation: For any two integers a, b such that $a \leq b$, let $[a{:}b]$ denote the set $\{a, a+1, \ldots, b\}$. Let $X_{[a:b]}$ denote the set $\{X_a, X_{a+1}, \ldots, X_b\}$. For any index set $\mathcal{I} = \{i_1, i_2, \ldots, i_n\}$, $X_{\mathcal{I}}$ denotes the set $\{X_{i_1}, X_{i_2}, \ldots, X_{i_n}\}$. For two vectors $\mathbf{A}$ and $\mathbf{B}$, $\mathbf{A} \perp\!\!\!\perp \mathbf{B}$ denotes that they are linearly independent. The notation $\mathbf{A}'$ denotes the transpose of $\mathbf{A}$, and $\mathbf{A}(i)$ denotes the i^{th} entry of $\mathbf{A}$. For an n-dimensional tensor $\mathbf{C}$, the notation $\mathbf{C}(i_1, i_2, \ldots, i_n)$ represents the entry at the corresponding position of $\mathbf{C}$. If $\mathbf{C}$ is a two-dimensional tensor, then it is a matrix and $\mathbf{C}(i_1, i_2)$ denotes the $(i_1, i_2)^{th}$ entry of matrix $\mathbf{C}$. The notation $(x)^+$ denotes $\max(x, 0)$. If A is a set of random variables, then by $H(A)$ we denote the joint entropy of those random variables. Mutual information between sets of random variables are similarly defined with the notation $I(A; B)$. The notation $\mathbf{e}_K(\theta)$ denotes the θ^{th} column of the $K \times K$ identity matrix.

II. Problem Statement: MB-XS-TPIR

Consider a database $\mathbf{W}$ comprised of $K = K_1 K_2 \cdots K_M$ messages, indexed as

$$\mathbf{W} = (\mathbf{W}(k_1, k_2, \ldots, k_M))_{k_1 \in [1\,:\,K_1], \ldots, k_M \in [1:K_M]}. \tag{1}$$

Each message consists of a stream of i.i.d. uniform bits. The *stream* of symbols implies that the message lengths are unbounded (a standard assumption in information theory). However, we are interested primarily in *bounded-latency* MB-XS-TPIR schemes, i.e., schemes that code over a bounded number of bits. For example, consider an encoder that accepts as input L symbols from $\mathbb{F}_q$ for each message, i.e., $L \log_2(q)$ bits of each message, and jointly encodes them. In order to jointly encode its inputs, the encoder must first wait to collect $L \log_2(q)$ bits of data for each message, thus introducing a coding delay, or latency. By bounded latency, we mean that L, q are $O(1)$ in the parameters $K_1, K_2, \ldots, K_M$. In other words, the number of bits that are jointly encoded by the MB-XS-TPIR scheme is bounded even as the number of messages approaches infinity. This assumption is important in practice, especially for streaming or dynamic data. To our knowledge, for all PIR settings where the asymptotic (large number of messages) capacity is known, it is achieved by bounded-latency schemes [26]. So we do not expect the bounded latency assumption to affect the asymptotic capacity of MB-XS-TPIR. But it will be a useful assumption for converse arguments for the special case of DB-TPIR (Double Blind T-PIR). Another issue worth clarifying is that even though L is bounded while the number of messages is allowed to be much larger, the downloads still dominate the communication cost because the same queries can be re-used repeatedly to download the unbounded desired message stream, L symbols at a time.

Under the bounded latency assumption, without loss of generality we will assume that each message has length L symbols. In q-ary units,

$$H(\mathbf{W}(k_1, k_2, \ldots, k_M)) = L, \quad \forall k_1 \in [1 : K_1], \ldots, k_M \in [1 : K_M], \tag{2}$$

$$H(\mathbf{W}) = \sum_{k_1 \in [1:K_1], \ldots, k_M \in [1\,:\,K_M]} H(\mathbf{W}(k_1, k_2, \ldots, k_M)) = K_1 K_2 \cdots K_M L. \tag{3}$$

The database $\mathbf{W}$ is stored at N distributed servers according to an X-secure storage scheme. Let the storage at the n^{th} server be denoted by $\mathbf{S}_n, n \in [1 : N]$. An X-secure storage scheme ensures that any set of up to X colluding servers cannot learn anything about the database $\mathbf{W}$.

$$\big[\text{X-Security}\big] \quad I(\mathbf{W}; \mathbf{S}_{\mathcal{X}}) = 0 \quad \forall \mathcal{X} \subset [1 : N], |\mathcal{X}| \leq X. \tag{4}$$

The setting $X = 0$ corresponds to replicated storage, where we set $\mathbf{S}_n = \mathbf{W}\ \forall n \in [1 : N]$.

There are M users. The user $m, m \in [1 : M]$ specifies the index θ_m which is uniform over $[1 : K_m]$. The M users jointly want to retrieve the message $\mathbf{W}(\theta_1, \theta_2, \ldots, \theta_M)$. The m^{th} user must keep its[4] index private against collusion among any set of up to T_m servers. Each user must also keep its index private against other users.

To this end, we assume for each $m \in [1 : M]$, user m has its own private randomness $\mathcal{Z}_m$. Note that $\mathcal{Z}_m$ is used to guarantee user m's T_m-privacy against any T_m colluding servers. The N servers share[5] common randomness $\widetilde{\mathcal{Z}}$ that is not available to

[4]The use of 'it' instead of 'he/she' for users reflects the motivating example of M-factor authentication, where different users may in fact be different inanimate devices owned by the same person.

[5]We need common randomness at the servers only to ensure perfect inter-user privacy, as in (9). Remarkably, *almost-perfect* inter-user privacy can be guaranteed (for large messages) even without common randomness at servers (see Corollary 1).

the users. The independence among these entities is formalized as follows.

$$H\big(\mathbf{S}_{[1:N]}, \widetilde{\mathcal{Z}}, (\theta_m)_{m\in[1:M]}, (\mathcal{Z}_m)_{m\in[1:M]}\big) = H\big(\mathbf{S}_{[1:N]}\big) + H\big(\widetilde{\mathcal{Z}}\big) + \sum_{m\in[1:M]} H(\theta_m) + \sum_{m\in[1:M]} H(\mathcal{Z}_m). \tag{5}$$

In order to retrieve the desired message, user m generates N queries $Q_1^{(m,\theta_m)}, Q_2^{(m,\theta_m)}, \ldots, Q_N^{(m,\theta_m)}$ based on its index θ_m and its private randomness $\mathcal{Z}_m$. Specifically,

$$H\Big(Q_{[1:N]}^{(m,\theta_m)}|\theta_m, \mathcal{Z}_m\Big) = 0 \quad \forall m \in [1:M]. \tag{6}$$

The corresponding queries from all M users, $(Q_n^{(m,\theta_m)})_{m\in[1:M]}$ are sent to the n^{th} server, for all $n \in [1:N]$. Upon receiving the queries, the n^{th} server generates its answer $A_n^{(\theta_1,\ldots,\theta_M)}$ as a function of the queries, the stored information and the server-side common randomness.

$$H\left(A_n^{(\theta_1,\ldots,\theta_M)}|\mathbf{S}_n, \left(Q_n^{(m,\theta_m)}\right)_{m\in[1:M]}, \widetilde{\mathcal{Z}}\right) = 0. \tag{7}$$

The privacy constraints consist of two parts.

1) (T_m)-Privacy. This means that any T_m or fewer servers have no knowledge about θ_m,

$$I\left(\theta_m; \left(Q_{\mathcal{T}}^{(i,\theta_i)}\right)_{i\in[1:M]}|\mathbf{S}_{\mathcal{T}}, \widetilde{\mathcal{Z}}\right) = 0, \quad \forall m \in [1:M], \mathcal{T} \subset [1:N], |\mathcal{T}| \le T_m. \tag{8}$$

2) Inter-user Privacy. This means that any user must learn nothing about other users' indices.

$$I\Big(\theta_{[1:M]\setminus\{m\}}; A_{[1:N]}^{(\theta_1,\ldots,\theta_M)}|\theta_m, \mathcal{Z}_m, \mathbf{W}(\theta_1,\ldots,\theta_M)\Big) = 0 \quad \forall m \in [1:M]. \tag{9}$$

With the answers from the N servers, each user must be able to recover the desired message.

$$\text{[Correctness]} \quad H\Big(\mathbf{W}(\theta_1,\ldots,\theta_M)|A_{[1:N]}^{(\theta_1,\ldots,\theta_M)}, \theta_m, \mathcal{Z}_m\Big) = 0 \quad \forall m \in [1:M]. \tag{10}$$

Recall that the rate of a PIR scheme is the number of bits of desired message that can be retrieved per bit of total download. Therefore, if D is the maximum (over all realizations of messages) number of q-ary symbols downloaded from all servers by a user, under an MB-XS-TPIR scheme that allows the user to retrieve L q-ary symbols of the desired message, then the rate of such a scheme is denoted as,

$$R = \frac{L}{D}. \tag{11}$$

The main contribution of this work is an achievable scheme for MB-XS-TPIR that is based on cross-subspace alignment, and achieves the rate $1-(X+T_1+\cdots+T_M)/N$, for arbitrary number of messages $K_1, K_2, \ldots, K_M$. Note that the scheme itself is not limited to asymptotic settings. Asymptotic settings will be of interest primarily for the purpose of testing the optimality of the scheme for significant special cases.

In order to introduce the scheme in a transparent setting, and to gain deeper insights into its optimality, we focus in particular on Double Blind T-PIR (DB-TPIR), which is obtained as a special case of MB-XS-TPIR by setting $M = 2, X = 0$. Given $q, L, N, K_1, K_2, T_1, T_2$ let us denote the supremum of rates achievable by any DB-TPIR scheme with these parameters as $R^*_{\text{DB-TPIR}}(q, L, N, K_1, K_2, T_1, T_2)$. Let us then define the capacity of DB-TPIR with parameters N, K_1, K_2, T_1, T_2 as

$$C_{\text{DB-TPIR}}(N, K_1, K_2, T_1, T_2) = \sup_{q,L} R^*_{\text{DB-TPIR}}(q, L, N, K_1, K_2, T_1, T_2). \tag{12}$$

Specifically, from the optimality perspective, we are interested in the asymptotic capacity of DB-TPIR as $K_1, K_2 \to \infty$. Under the bounded latency ($b.l.$) constraint, this asymptotic capacity is defined as

$$C^{\infty,b.l.}_{\text{DB-TPIR}}(N, T_1, T_2) \triangleq \sup_{q,L} \lim_{K_1,K_2\to\infty} R^*_{\text{DB-TPIR}}(q, L, N, K_1, K_2, T_1, T_2). \tag{13}$$

In plain words, $C^{\infty,b.l.}_{\text{DB-TPIR}}(N, T_1, T_2)$ is the highest rate possible for any DB-TPIR scheme when the number of messages is much larger than the number of bits of each message that are jointly encoded by the scheme.

Remark 1: For a double sequence $s(K_1, K_2)$, the notation $\lim_{K_1,K_2\to\infty} s(K_1, K_2) = a$ means that $\forall \epsilon > 0, \exists \kappa = \kappa(\epsilon)$ such that $|s(K_1, K_2) - a| < \epsilon \ \forall K_1, K_2 \ge \kappa$. (see [69, Definition 2.1]). It follows from [69, Th. 4.2] that the double limit $\lim_{K_1,K_2\to\infty} R^*_{\text{DB-TPIR}}$ exists. This is because $R^*_{\text{DB-TPIR}}$ is a decreasing sequence in each of K_1 and K_2 parameters individually (because any scheme that works with more messages also works with fewer messages), and is bounded below by zero. It also follows from in [69, Th. 4.2] that

$$\lim_{K_1,K_2\to\infty} R^*_{\text{DB-TPIR}} = \lim_{K_1\to\infty} \lim_{K_2\to\infty} R^*_{\text{DB-TPIR}}. \tag{14}$$

Remark 2: Note that the bounded-latency constraint affects the order in which the supremum is taken over message size parameters (q, L) versus the limit on the number of messages (K_1, K_2). Without the bounded latency constraint, the asymptotic capacity as the number of messages approaches infinity, would be defined as

$$C^{\infty}_{\text{DB-TPIR}}(N, T_1, T_2) = \lim_{K_1,K_2\to\infty} \sup_{q,L} R^*_{\text{DB-TPIR}}(q, L, N, K_1, K_2, T_1, T_2) = \lim_{K_1,K_2\to\infty} C_{\text{DB-TPIR}}(N, K_1, K_2, T_1, T_2). \tag{15}$$

Comparing (15) with (13), we note the key difference is that in (15), the supremum over message size (q, L) allows message sizes to approach infinity for a fixed number of messages, and only then the number of messages approaches infinity, whereas in (13) it is the number of messages (K_1, K_2) that approaches infinity first for a given message size (q, L are bounded, i.e., $O(1)$ in K_1, K_2), and only then the size of the message is allowed to grow. In a nutshell, (15) corresponds to asymptotic settings with $q^L \gg K_1, K_2$, while (13) corresponds to asymptotic settings with $q^L \ll K_1, K_2$, thus prioritizing coding latency.

III. Results

We begin with the asymptotic capacity characterization of DB-TPIR under the bounded-latency constraint.

Theorem 1: The asymptotic capacity of DB-TPIR subject to bounded-latency constraint is

$$C^{\infty,b.l.}_{\text{DB-TPIR}}(N,T_1,T_2)=\left(1-\left(\frac{T_1+T_2}{N}\right)\right)^+. \quad (16)$$

The proof of Theorem 1 is presented in Section IV. Notably, the achievability of the rate expression that appears on the RHS of (16) needs neither the bounded-latency assumption, nor the asymptotic setting. Both of those are needed primarily for the converse argument.

Next we examine the need for common randomness across servers. Common randomness is needed across servers primarily to preserve inter-user privacy, i.e., to keep each user's index private from other users. While in the absence of common randomness, our achievable scheme does not preserve inter-user privacy *perfectly*, it is remarkable that the scheme manages to preserve inter-user privacy *almost-perfectly* for large alphabet. In other words, the amount of information leaked to a user about the other user's index, is vanishingly small as $q\to\infty$. Corollary 1 highlights this observation by studying explicitly the case $T_1=T_2=1, K_1=K_2=K$.

Corollary 1: For the DB-TPIR scheme proposed in Section IV-B, let $B^{(\theta_1,\theta_2)}_{[1:N]}$ denote the answers generated by the N servers after eliminating common randomness between servers (setting all symbols associated with $\widetilde{\mathcal{Z}}$ to zero in our achievable scheme for DB-TPIR). For $T_1=T_2=1, K_1=K_2=K$ where K is a fixed positive integer, and for any $\epsilon>0$, there exists $q_0>0$ s.t. when $q\geq q_0$ (q is the size of the finite field $\mathbb{F}_q$),

$$I\Big(\theta_2; B^{(\theta_1,\theta_2)}_{[1:N]}|\theta_1,\mathcal{Z}_1,\mathbf{W}(\theta_1,\theta_2)\Big)\leq\epsilon, \quad (17)$$

$$I\Big(\theta_1; B^{(\theta_1,\theta_2)}_{[1:N]}|\theta_2,\mathcal{Z}_2,\mathbf{W}(\theta_1,\theta_2)\Big)\leq\epsilon. \quad (18)$$

The proof of Corollary 1 appears in Appendix A.

Our final result generalizes the achievable scheme from DB-TPIR to MB-XS-TPIR based on a tensor-product extension of cross-subspace alignment. The achievable rate of the general scheme is presented in the following theorem.

Theorem 2: For the MB-XS-TPIR problem defined in Section II, the following rate is achievable regardless of the number of messages $K_1,K_2,\ldots,K_M$.

$$R_{\text{MB-XS-TPIR}}=1-\frac{X+T_1+T_2+\cdots+T_M}{N}. \quad (19)$$

Intuitively, this rate expression indicates that with this scheme one symbol is downloaded from each server, and from those N symbols each user is able to recover $L=N-(X+T_1+T_2+\cdots+T_M)$ symbols of the desired message $\mathbf{W}(\theta_1,\theta_2,\ldots,\theta_M)$, while the interference is aligned within $X+T_1+T_2+\cdots+T_M$ dimensions. Theorem 2 is proved in Section V.

Corollary 2: Let us denote the supremum of achievable rates of MB-XS-TPIR (over all valid MB-XS-TPIR schemes) for fixed parameters $q,L,N,X,K_1,\ldots,K_M,T_1,\ldots,T_M$ as $R^*_{\text{MB-XS-TPIR}}$. Further, let us define the capacity of MB-XS-TPIR as $C_{\text{MB-XS-TPIR}}=\sup_{q,L}R^*_{\text{MB-XS-TPIR}}$. Then we have the following bounds,

$$\begin{aligned}1-\frac{X+T_1+T_2+\cdots+T_M}{N}&\leq C_{\text{MB-XS-TPIR}}\\&\leq\min\left(\frac{1-\frac{T_1+X}{N}}{1-\left(\frac{T_1}{N-X}\right)^{K_1}},\ldots,\frac{1-\frac{T_M+X}{N}}{1-\left(\frac{T_M}{N-X}\right)^{K_M}}\right). \quad (20)\end{aligned}$$

The proof of Corollary 2 appears in Appendix B. The lower bound in (20) follows directly from the proof of achievability of Theorem 2. The upper bound in (20) is obtained by noting that MB-XS-TPIR schemes automatically yield XS-TPIR schemes. By setting $M=2$ and $X=0$, the capacity of DB-TPIR is bounded as

$$\begin{aligned}1-\frac{T_1+T_2}{N}&\leq C_{\text{DB-TPIR}}\\&\leq\min\left(\frac{1-T_1/N}{1-(T_1/N)^{K_1}},\frac{1-T_2/N}{1-(T_2/N)^{K_2}}\right). \quad (21)\end{aligned}$$

IV. Asymptotic Capacity of DB-TPIR

This section is devoted to the proof of Theorem 1.

A. Theorem 1: Converse

Let us find an upper bound on the capacity of DB-TPIR by noting a relationship between DB-TPIR and X-secure T-private information retrieval (XS-TPIR) [29]. Recall that XS-TPIR is a special case of MB-XS-TPIR obtained by setting $M=1$. The capacity of XS-TPIR with N distributed servers, K messages, X-secure data storage, and T-private queries is denoted as $C_{\text{XS-TPIR}}(N,K,X,T)$. Recall that the asymptotic capacity of XS-TPIR (as $K\to\infty$) is shown in [29] to be $C^{\infty}_{\text{XS-TPIR}}(N,X,T)=(1-\frac{X+T}{N})^+$.

We will need the following lemma.

Lemma 1: Let $R^*_{\text{DB-TPIR}}(q,L,N,K_1,K_2,T_1,T_2)$ denote the supremum of rates achievable by any DB-TPIR scheme for the parameters q,L,N,K_1,K_2,T_1,T_2 as defined in Section II. Then for $K_2=q^{LK_1}$, we have,

$$\begin{aligned}&R^*_{\text{DB-TPIR}}\big(q,L,N,K_1,K_2=q^{LK_1},T_1,T_2\big)\\&\quad\leq C_{\text{XS-TPIR}}(N,K=K_1,X=T_2,T=T_1). \quad (22)\end{aligned}$$

Proof: Consider a $K_1\times K_2$ matrix $\dot{\mathbf{W}}$ whose elements are from $\mathbb{F}_q^L$. The K_2 column vectors are all distinct and, say, arranged in lexicographic order. Since $K_2=q^{LK_1}$, the column vectors of the matrix include all q^{LK_1} possible realizations of $K_1\times 1$ vectors over $\mathbb{F}_q^L$, and $\dot{\mathbf{W}}$ is uniquely specified. We claim that any construction of a DB-TPIR scheme for the parameter values specified on the LHS of (22), when applied with the particular realization of the database $\mathbf{W}=\dot{\mathbf{W}}$, yields an XS-TPIR scheme with the parameters specified on the RHS of (22).

Let us describe this XS-TPIR scheme. In this XS-TPIR scheme the user corresponds to User 1 of the DB-TPIR scheme. Each Server n stores only $Q_n^{(2,\theta_2)}$. Note that $\dot{\mathbf{W}}$ is

a constant matrix known to everyone, whose θ_2^{th} column specifies the realizations of the K_1 i.i.d. messages (one of which is desired by the user), each comprised of L uniformly random i.i.d. symbols from $\mathbb{F}_q$. Since θ_2 is T_2-private according to the DB-TPIR construction, this constitutes $X = T_2$-secure storage of the K_1 messages. Furthermore, based on the T_1-private queries, $Q_n^{(1,\theta_1)}$, provided by the user, each server is able to respond as in the DB-TPIR scheme (because $Q_n^{(2,\theta_2)}$ is already known to Server n), and the DB-TPIR construction guarantees that the desired message $\dot{\mathbf{W}}(\theta_1, \theta_2)$ is correctly retrieved. Finally, since the rate of an XS-TPIR scheme cannot be higher than the capacity of XS-TPIR, the proof of Lemma 1 is complete. ■

Remark 3: The XS-TPIR scheme that we obtain from the DB-TPIR scheme described above, allows common randomness between servers. While the original formulation of XS-TPIR in [29] does not explicitly allow common randomness, it is readily verified that server-side common randomness can be included in the storage of each server in the model of [29], and the asymptotic capacity result still holds.

Proof of Converse of Theorem 1: Note that although the proof of Lemma 1 requires the condition that $K_1 = q^{LK_2}$, Theorem 1 must hold as long as both K_1 and K_2 grow unbounded, regardless of their growth rates. For this we will utilize (14) as follows.

$$
\begin{aligned}
&C^{\infty,b.l.}_{\text{DB-TPIR}}(N, T_1, T_2)\\
&\quad= \sup_{q,L} \lim_{K_1,K_2\to\infty} R^*_{\text{DB-TPIR}}(q, L, N, K_1, K_2, T_1, T_2) && (23)\\
&\quad= \sup_{q,L} \lim_{K_1\to\infty} \left(\lim_{K_2\to\infty} R^*_{\text{DB-TPIR}}(q, L, N, K_1, K_2, T_1, T_2) \right) && (24)\\
&\quad\le \sup_{q,L} \lim_{K_1\to\infty} \left(\lim_{K_2\to\infty} R^*_{\text{DB-TPIR}}\left(q, L, N, \log_{q^L}(K_1), K_2, T_1, T_2\right) \right) && (25)\\
&\quad\le \sup_{q,L} \lim_{K_1\to\infty} R^*_{\text{DB-TPIR}}\left(q, L, N, \log_{q^L}(K_1), K_1, T_1, T_2\right) && (26)\\
&\quad\le \sup_{q,L} \lim_{K_1\to\infty} C_{\text{XS-TPIR}}\left(N, K = \log_{q^L}(K_1), X = T_2, T = T_1\right) && (27)\\
&\quad= \sup_{q,L} \lim_{K\to\infty} C_{\text{XS-TPIR}}(N, K, X = T_2, T = T_1) && (28)\\
&\quad= \lim_{K\to\infty} C_{\text{XS-TPIR}}(N, K, X = T_2, T = T_1) && (29)\\
&\quad= \begin{cases} 1 - \left(\frac{T_1+T_2}{N}\right), & N > T_1 + T_2 \\ 0, & N \le T_1 + T_2. \end{cases} && (30)
\end{aligned}
$$

The first step, (24), follows directly from (14). In (25) we used the fact that reducing the number of messages cannot hurt the rate (because the original scheme can still be used with fewer messages). The next step, (26) follows because when $K_2 \to \infty$, K_1 is viewed as a constant which is less than K_2 and reducing the number of messages cannot hurt the rate. For (27) we used Lemma 1. The next step, (28) follows because for fixed q, L, and $K = \log_{q^L}(K_1)$, the condition that $K_1 \to \infty$ is equivalent to the condition that $K \to \infty$. Next, (29) follows because the capacity expression is not a function of q or L. Finally, the asymptotic capacity characterization for XS-TPIR from [29] is used for (30). Thus, the proof of the converse part of Theorem 1 is complete. ■

B. Theorem 1: Achievability

In this section, we prove the achievability of Theorem 1 by constructing a scheme based on Cross Subspace Alignment (CSA) Codes [61], that can achieve the rate $(1-(T_1+T_2)/N)^+$ for arbitrary N, K_1, K_2, T_1, T_2. We will focus only on the non-trivial case, $N > T_1 + T_2$. Throughout this scheme we set,

$$L = N - (T_1 + T_2). \tag{31}$$

Each message $\mathbf{W}(i,j), i \in [1{:}K_1], j \in [1{:}K_2]$ consists of L symbols from finite field $\mathbb{F}_q$, denoted as $\mathbf{W}(i,j) = (\mathbf{W}(i,j)^{(1)}, \mathbf{W}(i,j)^{(2)}, \ldots, \mathbf{W}(i,j)^{(L)})$. For the scheme we will need the following $L+N$ distinct constants from $\mathbb{F}_q$,

$$f_1, f_2, \ldots, f_L, \alpha_1, \alpha_2, \ldots, \alpha_N \tag{32}$$

that are known to all N servers and the 2 users. Note that this implies that $q \ge L + N$.

Let us split the messages $\mathbf{W}$ into L matrices $(\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \ldots, \mathbf{W}^{(L)})$ so that $\mathbf{W}^{(l)}, l \in [1{:}L]$ contains the l^{th} symbol of each message. Specifically,

$$\mathbf{W}^{(l)} = \begin{bmatrix} \mathbf{W}(1,1)^{(l)} & \mathbf{W}(1,2)^{(l)} & \cdots & \mathbf{W}(1,K_2)^{(l)} \\ \mathbf{W}(2,1)^{(l)} & \mathbf{W}(2,2)^{(l)} & \cdots & \mathbf{W}(2,K_2)^{(l)} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{W}(K_1,1)^{(l)} & \mathbf{W}(K_1,2)^{(l)} & \cdots & \mathbf{W}(K_1,K_2)^{(l)} \end{bmatrix}. \tag{33}$$

Note that we write equivalently $\mathbf{W}^{(l)}(\theta_1, \theta_2) = \mathbf{W}(\theta_1, \theta_2)^{(l)}$.

Recall that $\mathbf{e}_K(\theta)$ is the θ^{th} column of the $K \times K$ identity matrix. The l^{th} symbol of $\mathbf{W}(\theta_1, \theta_2)$ can be expressed as

$$\mathbf{W}(\theta_1, \theta_2)^{(l)} = \mathbf{e}_{K_1}(\theta_1)' \mathbf{W}^{(l)} \mathbf{e}_{K_2}(\theta_2). \tag{34}$$

Note here $\mathbf{e}_{K_1}(\theta_1)'\mathbf{W}^{(l)}$ is the θ_1^{th} row of matrix $\mathbf{W}^{(l)}$. The inner product of θ_1^{th} row with $\mathbf{e}_{K_2}(\theta_2)$ is the entry at the θ_2^{th} column of this row, i.e., $\mathbf{W}(\theta_1, \theta_2)^{(l)}$. The proposed scheme will enable the 2 users to retrieve $\mathbf{W}(\theta_1, \theta_2)^{(l)}$ $\forall l \in [1{:}L]$, thus, retrieving $\mathbf{W}(\theta_1, \theta_2)$.

The private randomness available to each user is specified as,

$$\mathcal{Z}_1 = \left\{ \mathbf{Z}_{1,t}^{(l)} \mid t \in [1 : T_1], l \in [1 : L] \right\}, \tag{35}$$

$$\mathcal{Z}_2 = \left\{ \mathbf{Z}_{2,t}^{(l)} \mid t \in [1 : T_2], l \in [1 : L] \right\}. \tag{36}$$

The random vectors $\mathbf{Z}_{1,t}^{(l)} \in \mathbb{F}_q^{K_1 \times 1}, \mathbf{Z}_{2,t}^{(l)} \in \mathbb{F}_q^{K_2 \times 1}$ have their elements drawn i.i.d. uniform from $\mathbb{F}_q$.

The query sent by user $m, m \in \{1, 2\}$ to the n^{th} server, $Q_n^{(m,\theta_m)}$ is constructed as $Q_n^{(m,\theta_m)} = (Q_{n,1}^{(m,\theta_m)}, Q_{n,2}^{(m,\theta_m)}, \ldots, Q_{n,L}^{(m,\theta_m)})$ where $\forall l \in [1 : L]$

$$Q_{n,l}^{(m,\theta_m)} = \mathbf{e}_{K_m}(\theta_m) + \sum_{t \in [1:T_m]} (f_l - \alpha_n)^t \mathbf{Z}_{m,t}^{(l)}. \tag{37}$$

Specifically, $Q_{n,l}^{(m,\theta_m)} \in \mathbb{F}_q^{K_m \times 1}$ can be viewed as the query from user m to request the l^{th} symbol of the wanted message. The T_m-privacy constraint is satisfied since $Q_{n,l}^{(m,\theta_m)}$ is the Shamir's secret sharing [1] of $\mathbf{e}_{K_m}(\theta_m)$. Up to T_m colluding servers can learn nothing about $\mathbf{e}_{K_m}(\theta_m)$, thus, learning nothing about θ_m.

Upon receiving queries from both users, the n^{th} server computes an intermediate result

$$B_n^{(\theta_1,\theta_2)} = \sum_{l\in[1:L]} \frac{1}{f_l-\alpha_n} Q_{n,l}^{(1,\theta_1)\prime}\mathbf{W}^{(l)}Q_{n,l}^{(2,\theta_2)} \tag{38}$$

$$= \frac{1}{f_1-\alpha_n}\underbrace{\mathbf{e}_{K_1}(\theta_1)'\mathbf{W}^{(1)}\mathbf{e}_{K_2}(\theta_2)}_{\mathbf{W}(\theta_1,\theta_2)^{(1)}} + \cdots + \frac{1}{f_L-\alpha_n}\underbrace{\mathbf{e}_{K_1}(\theta_1)'\mathbf{W}^{(L)}\mathbf{e}_{K_2}(\theta_2)}_{\mathbf{W}(\theta_1,\theta_2)^{(L)}} + I_0+\alpha_n I_1+\cdots+\alpha_n^{T_1+T_2-1}I_{T_1+T_2-1}. \tag{39}$$

From (38) to (39), distributive law is used. Note that (39) can be viewed as a polynomial of α_n. The coefficients of the first L terms are the L symbols of the desired message. $I_i, i \in [0:T_1+T_2-1]$ stands for the remaining (interference) terms that are generated by the product of the matrices in (38). The highest power of α_n is T_1+T_2-1 and can be found from

$$\sum_{l\in[1:L]} (f_l-\alpha_n)^{T_1+T_2-1}\mathbf{Z}_{1,T_1}^{(l)\prime}\mathbf{W}^{(l)}\mathbf{Z}_{2,T_2}^{(l)}.$$

Note that the interference terms of (39), except the one of the highest order, may contain some information of the index specified by a user. For example, I_0 contains

$$\frac{1}{f_l-\alpha_n}\mathbf{e}_{K_1}(\theta_1)'\mathbf{W}^{(l)}(f_l-\alpha_n)\mathbf{Z}_{2,1}^{(l)} = \mathbf{e}_{K_1}(\theta_1)'\mathbf{W}^{(l)}\mathbf{Z}_{2,1}^{(l)},$$

which means that User 2 may get some information about the index θ_1 specified by User 1 from the interference terms. To protect against this leakage of information, server n will add noise drawn from the common randomness that is shared by all servers. The common randomness shared among N servers is specified as,

$$\widetilde{\mathcal{Z}} = \left\{\widetilde{Z}_i \mid i\in[0:T_1+T_2-1]\right\}, \tag{40}$$

where $(\widetilde{Z}_i)_{i\in[0:T_1+T_2-1]}$ are T_1+T_2 random variables that are i.i.d. uniform over $\mathbb{F}_q$. Server n will add the polynomial

$$\widetilde{Z}(\alpha_n) = \widetilde{Z}_0+\alpha_n\widetilde{Z}_1+\cdots+\alpha_n^{T_1+T_2-1}\widetilde{Z}_{T_1+T_2-1} \tag{41}$$

to the intermediate result $B_n^{(\theta_1,\theta_2)}$ to generate its answer $A_n^{(\theta_1,\theta_2)}$. This is the answer sent to both users.

$$A_n^{(\theta_1,\theta_2)} = B_n^{(\theta_1,\theta_2)} + \widetilde{Z}(\alpha) \tag{42}$$

$$= \frac{1}{f_1-\alpha_n}\mathbf{W}(\theta_1,\theta_2)^{(1)} + \cdots + \frac{1}{f_L-\alpha_n}\mathbf{W}(\theta_1,\theta_2)^{(L)} + \underbrace{\left(I_0+\widetilde{Z}_0\right)}_{J_0} + \cdots + \alpha_n^{T_1+T_2-1} \times \underbrace{\left(I_{T_1+T_2-1}+\widetilde{Z}_{T_1+T_2-1}\right)}_{J_{T_1+T_2-1}}. \tag{43}$$

Rewriting (43) in matrix multiplication form, we have (44) shown at the bottom of the page. The matrix $\mathbf{C}$ is a Cauchy-Vandermonde matrix of size $N\times N$ since $N = L+T_1+T_2$. Since $f_l, l\in[1:L]$, $\alpha_n, n\in[1:N]$ are $L+N$ distinct elements of $\mathbb{F}_q$, according to [70], $\mathbf{C}$ is invertible in $\mathbb{F}_q$. Thus, the answers from all the N servers form an invertible function of $\mathbf{W}(\theta_1,\theta_2), J_0,\ldots,J_{T_1+T_2-1}$. In other words, the correctness constraint is satisfied.

Let us consider the inter-user privacy. Without loss of generality, let us consider User 1. We have

$$I\left(\theta_2; A_{[1:N]}^{(\theta_1,\theta_2)}|\theta_1, \mathcal{Z}_1, \mathbf{W}(\theta_1,\theta_2)\right) \tag{45}$$

$$= I\left(\theta_2; \mathbf{W}(\theta_1,\theta_2), J_{[0:T_1+T_2-1]}|\theta_1, \mathcal{Z}_1, \mathbf{W}(\theta_1,\theta_2)\right) \tag{46}$$

$$= I\left(\theta_2; J_{[0:T_1+T_2-1]}|\theta_1, \mathcal{Z}_1, \mathbf{W}(\theta_1,\theta_2)\right) = 0. \tag{47}$$

Equation (47) comes from the fact that $J_{[0:T_1+T_2-1]}$ are protected by T_1+T_2 random symbols shared among servers, which are uniformly i.i.d. over $\mathbb{F}_q$ and are independent of all other terms in (47).

Finally, note that since $L = N-(T_1+T_2)$ symbols of the desired message are retrieved from a total of N downloaded symbols from all N servers, the rate of this scheme is $L/N = 1-(T_1+T_2)/N$.

C. Examples for Illustration

1) $L=1, T_1=T_2=1$ With $N=3$ Servers: Since $L=1, T=1$, we neglect the l,t on superscripts or subscripts of all symbols. The queries from the 2 users are listed as follows.

Server 'n'	
$Q_n^{(1,\theta_1)}$	$\mathbf{e}_{K_1}(\theta_1)+(f_1-\alpha_n)\mathbf{Z}_1$
$Q_n^{(2,\theta_2)}$	$\mathbf{e}_{K_2}(\theta_2)+(f_1-\alpha_n)\mathbf{Z}_2$

The intermediate result is computed as

$$B_n^{(\theta_1,\theta_2)} = \frac{1}{f_1-\alpha_n}Q_n^{(1,\theta_1)\prime}\mathbf{W}Q_n^{(2,\theta_2)} = \frac{1}{f_1-\alpha_n}\cdot\left(\mathbf{e}_{K_1}(\theta_1)'+(f_1-\alpha_n)\mathbf{Z}_1'\right)\cdot\mathbf{W} \times\left(\mathbf{e}_{K_2}(\theta_2)+(f_1-\alpha_n)\mathbf{Z}_2\right) = \frac{1}{f_1-\alpha_n}\mathbf{e}_{K_1}(\theta_1)'\mathbf{W}\mathbf{e}_{K_2}(\theta_2)$$

$$\begin{bmatrix} A_1^{(\theta_1,\theta_2)} \\ A_2^{(\theta_1,\theta_2)} \\ \cdots \\ A_N^{(\theta_1,\theta_2)} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{1}{f_1-\alpha_1} & \frac{1}{f_2-\alpha_1} & \cdots & \frac{1}{f_L-\alpha_1} & 1 & \alpha_1 & \cdots & \alpha_1^{T_1+T_2-1} \\ \frac{1}{f_1-\alpha_2} & \frac{1}{f_2-\alpha_2} & \cdots & \frac{1}{f_L-\alpha_2} & 1 & \alpha_2 & \cdots & \alpha_2^{T_1+T_2-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{f_1-\alpha_N} & \frac{1}{f_2-\alpha_N} & \cdots & \frac{1}{f_L-\alpha_N} & 1 & \alpha_N & \cdots & \alpha_N^{T_1+T_2-1} \end{bmatrix}}_{\mathbf{C}} \begin{bmatrix} \mathbf{W}(\theta_1,\theta_2)^{(1)} \\ \mathbf{W}(\theta_1,\theta_2)^{(2)} \\ \vdots \\ \mathbf{W}(\theta_1,\theta_2)^{(L)} \\ J_0 \\ \vdots \\ J_{T_1+T_2-1} \end{bmatrix} \tag{44}$$

$$
\begin{aligned}
&+ \left(\mathbf{Z}_1'\mathbf{W}\mathbf{e}_{K_2}(\theta_2) + \mathbf{e}_{K_1}(\theta_1)'\mathbf{W}\mathbf{Z}_2\right) \\
&+ (f_1 - \alpha_n)\mathbf{Z}_1'\mathbf{W}\mathbf{Z}_2 \\
&= \frac{1}{f_1 - \alpha_n}\mathbf{W}(\theta_1, \theta_2) \\
&+ \underbrace{\left(\mathbf{Z}_1'\mathbf{W}\mathbf{e}_{K_2}(\theta_2) + \mathbf{e}_{K_1}(\theta_1)'\mathbf{W}\mathbf{Z}_2 + f\mathbf{Z}_1'\mathbf{W}\mathbf{Z}_2\right)}_{I_0} \\
&+ \alpha_n \underbrace{\left(-\mathbf{Z}_1'\mathbf{W}\mathbf{Z}_2\right)}_{I_1}.
\end{aligned}
$$

The answer from the server is

$$
\begin{aligned}
A_n^{(\theta_1,\theta_2)} &= B_n^{(\theta_1,\theta_2)} + \widetilde{Z}_0 + \alpha_n\widetilde{Z}_1 \\
&= \frac{1}{f_1 - \alpha_n}\mathbf{W}(\theta_1, \theta_2) + J_0 + \alpha_n J_1.
\end{aligned}
$$

Writing in matrix form, the answers from $N = 3$ servers are

$$
\begin{bmatrix} A_1^{(\theta_1,\theta_2)} \\ A_2^{(\theta_1,\theta_2)} \\ A_3^{(\theta_1,\theta_2)} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{1}{f_1-\alpha_1} & 1 & \alpha_1 \\ \frac{1}{f_1-\alpha_2} & 1 & \alpha_2 \\ \frac{1}{f_1-\alpha_3} & 1 & \alpha_3 \end{bmatrix}}_{\mathbf{C}} \begin{bmatrix} \mathbf{W}(\theta_1, \theta_2) \\ J_0 \\ J_1 \end{bmatrix}.
$$

The desired message is retrieved by inverting the matrix $\mathbf{C}$. Since $L = N - (T_1 + T_2) = 1$ symbol of the desired message is retrieved from a total of $N = 3$ downloaded symbols from all 3 servers, the rate of the scheme is $L/N = 1/3$.

2) $L = 2, T_1 = 1, T_2 = 2$ With $N = 5$ Servers: The queries from the 2 users are listed as follows.

	Server 'n'
$Q_{n,1}^{(1,\theta_1)}$	$\mathbf{e}_{K_1}(\theta_1) + (f_1 - \alpha_n)\mathbf{Z}_{1,1}^{(1)}$
$Q_{n,2}^{(1,\theta_1)}$	$\mathbf{e}_{K_1}(\theta_1) + (f_2 - \alpha_n)\mathbf{Z}_{1,1}^{(2)}$
$Q_{n,1}^{(2,\theta_2)}$	$\mathbf{e}_{K_2}(\theta_2) + (f_1 - \alpha_n)\mathbf{Z}_{2,1}^{(1)} + (f_1 - \alpha_n)^2\mathbf{Z}_{2,2}^{(1)}$
$Q_{n,2}^{(2,\theta_2)}$	$\mathbf{e}_{K_2}(\theta_2) + (f_2 - \alpha_n)\mathbf{Z}_{2,1}^{(2)} + (f_2 - \alpha_n)^2\mathbf{Z}_{2,2}^{(2)}$

The intermediate result is

$$
\begin{aligned}
B_n^{(\theta_1,\theta_2)} &= \frac{1}{f_1 - \alpha_n}Q_{n,1}^{(1,\theta_1)'}\mathbf{W}^{(1)}Q_{n,1}^{(2,\theta_2)} \\
&\quad + \frac{1}{f_2 - \alpha_n}Q_{n,2}^{(1,\theta_1)'}\mathbf{W}^{(2)}Q_{n,2}^{(2,\theta_2)} \\
&= \frac{1}{f_1 - \alpha_n}\mathbf{W}(\theta_1, \theta_2)^{(1)} + \frac{1}{f_2 - \alpha_n}\mathbf{W}(\theta_1, \theta_2)^{(2)} \\
&\quad + I_0 + \cdots + \alpha_n^2 I_2.
\end{aligned}
$$

The answer is

$$
\begin{aligned}
A_n^{(\theta_1,\theta_2)} &= \frac{1}{f_1 - \alpha_n}\mathbf{W}(\theta_1, \theta_2)^{(1)} + \frac{1}{f_2 - \alpha_n}\mathbf{W}(\theta_1, \theta_2)^{(2)} \\
&\quad + \underbrace{\left(I_0 + \widetilde{Z}_0\right)}_{J_0} + \cdots + \alpha_n^2 \underbrace{\left(I_2 + \widetilde{Z}_2\right)}_{J_2}.
\end{aligned}
$$

Writing in matrix form, the answers from $N = 5$ servers are

$$
\begin{bmatrix} A_1^{(\theta_1,\theta_2)} \\ A_2^{(\theta_1,\theta_2)} \\ A_3^{(\theta_1,\theta_2)} \\ A_4^{(\theta_1,\theta_2)} \\ A_5^{(\theta_1,\theta_2)} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{1}{f_1-\alpha_1} & \frac{1}{f_2-\alpha_1} & 1 & \alpha_1 & \alpha_1^2 \\ \frac{1}{f_1-\alpha_2} & \frac{1}{f_2-\alpha_2} & 1 & \alpha_2 & \alpha_2^2 \\ \frac{1}{f_1-\alpha_3} & \frac{1}{f_2-\alpha_3} & 1 & \alpha_3 & \alpha_3^2 \\ \frac{1}{f_1-\alpha_4} & \frac{1}{f_2-\alpha_4} & 1 & \alpha_4 & \alpha_4^2 \\ \frac{1}{f_1-\alpha_5} & \frac{1}{f_2-\alpha_5} & 1 & \alpha_5 & \alpha_5^2 \end{bmatrix}}_{\mathbf{C}} \begin{bmatrix} \mathbf{W}(\theta_1, \theta_2)^{(1)} \\ \mathbf{W}(\theta_1, \theta_2)^{(2)} \\ J_0 \\ J_1 \\ J_2 \end{bmatrix}.
$$

Evidently, the rate achieved is $L/N = 2/5$ in this case.

V. M-Way Blind X-Secure T-Private Information Retrieval

In this section, we propose a scheme that solves the generalized problem: M-way blind X-secure T-private information retrieval (MB-XS-TPIR). The rate achieved by this scheme is $R = 1 - (X + T_1 + \cdots + T_M)/N$.

MB-XS-TPIR has been formalized in Section II. In brief, MB-XS-TPIR enables M users who independently specify M indices $\theta_1, \ldots, \theta_M$ (θ_m is specified by user m) to retrieve a message $\mathbf{W}(\theta_1, \ldots, \theta_M)$ from a database $\mathbf{W}$ which is X-securely stored at N distributed servers, with (T_m)-Privacy and User-User Privacy constraints satisfied.

The MB-XS-TPIR scheme proposed in this section is still based on Cross Subspace Alignment (CSA) and is a natural extension of the DB-TPIR scheme. The main difference is that in this case, the database $\mathbf{W}$ is an M-dimensional tensor instead of a 2-dimensional matrix in DB-TPIR.

A. Brief Review of Tensors

Let us briefly review the key properties of tensors that we will need. Specifically, an M-dimensional tensor is an M-dimensional array. For instance a 2-dimensional tensor is a matrix, and a 3-dimensional tensor is a cuboid made up of several matrices. Each dimension of a tensor is called a mode. The m^{th} dimension is called mode-m. The tensor operation we mainly need is the operation called *mode-m tensor vector multiplication*. Readers can refer to [71, Ch. 3, Sec. 3.1.2] for more details.

Definition 1 (Mode-m Tensor Vector Multiplication): The mode-m multiplication of a tensor $\mathbf{A} \in \mathbb{F}_q^{K_1 \times K_2 \times \cdots \times K_M}$ with a column vector $\mathbf{b} \in \mathbb{F}_q^{K_m \times 1}$ results in the tensor,

$$
\mathbf{C} = \mathbf{A} \times_m \mathbf{b}, \tag{48}
$$

where $\mathbf{C} \in \mathbb{F}_q^{K_1 \times \cdots \times K_{m-1} \times 1 \times K_{m+1} \times \cdots \times K_M}$, and each element of $\mathbf{C}$ is specified as

$$
\begin{aligned}
&\mathbf{C}(k_1, \ldots, k_{m-1}, 1, k_{m+1}, \ldots, k_M) \\
&\quad = \sum_{k_m \in [1:K_m]} \mathbf{A}(k_1, \ldots, k_M) \cdot \mathbf{b}(k_m).
\end{aligned} \tag{49}
$$

Note that this operation is a multilinear operation, so distributive law applies to this operation.

B. General MB-XS-TPIR Scheme

Before formally presenting our MB-XS-TPIR solution, let us briefly explain at a high level how our solution translates into the problem of secure distributed tensor product computation. For our solution, we first arrange the data into L tensors $\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L)}$, where $\mathbf{W}^{(l)} \in \mathbb{F}_q^{K_1 \times K_2 \cdots \times K_M}, l \in [1:L]$ is comprised of the l^{th} symbol of each of the $K_1 K_2 \cdots K_M$ messages. The tensorized data is secret shared among the N servers as $(\mathbf{S}_n^{(1)}, \ldots, \mathbf{S}_n^{(L)})_{n \in [1:N]}$ to guarantee X-security. Next, the M vectors $\mathbf{e}_{K_1}(\theta_1), \ldots, \mathbf{e}_{K_M}(\theta_M)$, corresponding to the indices specified by the M users, are secret-shared among the N servers in the form of the queries $(Q_n^{(1,\theta_1)}, \ldots, Q_n^{(M,\theta_M)})_{n \in [1:N]}$

to retrieve the desired message. $(Q_n^{(m,\theta_m)})_{n\in[1:N]}$ is the secret-sharing of the query from the m^{th} user that ensures T_m privacy. Most importantly, with this construction of queries and tensorized data, retrieving the desired message corresponds to retrieving tensor products of the privatized queries and secured data. From this point on, the achievability scheme for MB-XS-TPIR can indeed be viewed as a secure coded tensor product computation, which is an multilinear operation with $M+1$ inputs, for which CSA codes [61] can be used. To optimize the download cost for MB-XS-TPIR, the parameters of the CSA codes are chosen as: $K_c = 1, \ell = N-(X+\sum_{m\in[1:M]} T_m)$. Note that the proposed scheme automatically recovers asymptotically optimal schemes for various special cases of MB-XS-TPIR, such as PIR, TPIR, XS-TPIR, etc. This further underscores the connection between various forms of PIR and coded distributed computing.

Now let us proceed to formally present our MB-XS-TPIR scheme. Throughout this scheme we set $L = N-(T_1+T_2+\cdots+T_M)-X$. Let $\mathbb{F}_q$ be a finite field with $q \geq L+N$ and let $f_1,\ldots,f_L,\alpha_1,\ldots,\alpha_N$ be $L+N$ distinct elements in $\mathbb{F}_q$. These $L+N$ elements are known to the N servers and M users.

The private randomness available at user m to keep its index θ_m T_m-private is

$$\mathcal{Z}_m = \left\{\mathbf{Z}_{m,t}^{(l)} \mid t\in[1:T_m], l\in[1:L]\right\} \quad \forall m\in[1:M], \quad (50)$$

where the column vectors $\mathbf{Z}_{m,t}^{(l)} \in \mathbb{F}_q^{K_m\times 1}$ have entries drawn i.i.d. uniform from $\mathbb{F}_q$.

For compact notation, we write $\sum T_m$ instead of $\sum_{m\in[1:M]} T_m$. The common randomness $\widetilde{\mathcal{Z}}$ shared among N servers for protecting inter-user privacy is specified as

$$\widetilde{\mathcal{Z}} = \left\{\widetilde{Z}_i \mid i\in\left[0:\sum T_m + X - 1\right]\right\}, \quad (51)$$

where $\widetilde{Z}_i, i\in[0:\sum T_m+X-1]$ are $\sum T_m+X$ random noise variables that are i.i.d. and uniform over $\mathbb{F}_q$.

To form X-secure storage of the data, let us introduce

$$\widehat{\mathcal{Z}} = \left\{\widehat{\mathbf{Z}}_{l,x} \mid x\in[1:X], l\in[1:L]\right\}, \quad (52)$$

which are independent uniform random noise tensors from $\mathbb{F}_q^{K_1\times\cdots\times K_M}$.

The database $\mathbf{W}$ can be split into L parts, each of which is an M-dimensional tensor. This partitioning is specified as

$$\mathbf{W} = \left(\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \ldots, \mathbf{W}^{(L)}\right), \quad \mathbf{W}^{(l)} \in \mathbb{F}_q^{K_1\times K_2\times\cdots\times K_M} \quad \forall l\in[1:L], \quad (53)$$

so that $\mathbf{W}^{(l)}$ contains the l^{th} symbol of every message.

The independence between the messages, indices, and noises is specified as

$$\begin{aligned} &H\left(\mathbf{W}, (\theta_m)_{m\in[1:M]}, (\mathcal{Z}_m)_{m\in[1:M]}, \widetilde{\mathcal{Z}}, \widehat{\mathcal{Z}}\right) \\ &= \sum_{l\in[1:L]} H\left(\mathbf{W}^{(l)}\right) + \sum_{m\in[1:M]} H(\theta_m) \\ &\quad + \sum_{m\in[1:M]} H(\mathcal{Z}_m) + H\left(\widetilde{\mathcal{Z}}\right) + H\left(\widehat{\mathcal{Z}}\right) \\ &= LK_1\cdots K_M + \sum_{m\in[1:M]} H(\theta_m) + \sum_{m\in[1:M]} LK_mT_m \\ &\quad + \sum_{m\in[1:M]} T_m + X + LK_1\cdots K_MX. \end{aligned} \quad (54)$$

To keep the database $\mathbf{W}$ X-secure, $\mathbf{W}$ is secret-shared among N servers. The n^{th} server holds the share $\mathbf{S}_n = (\mathbf{S}_n^{(1)},\ldots,\mathbf{S}_n^{(L)})$ where

$$\mathbf{S}_n^{(l)} = \mathbf{W}^{(l)} + \sum_{x\in[1:X]} (f_l-\alpha_n)^x \widehat{\mathbf{Z}}_{l,x}. \quad (55)$$

Note that $\mathbf{e}_K(\theta)$ is the θ^{th} column of the $K\times K$ identity matrix. With the tensor vector multiplication defined above, the desired message can be written as

$$\begin{aligned} \mathbf{W}(\theta_1,\ldots,\theta_M) &= \left(\mathbf{W}^{(l)}(\theta_1,\ldots,\theta_M)\right)_{l\in[1:L]} \\ &= \Big(\mathbf{W}^{(l)} \times_1 \mathbf{e}_{K_1}(\theta_1) \times_2 \mathbf{e}_{K_2}(\theta_2) \times_3 \cdots \\ &\qquad \times_M \mathbf{e}_{K_M}(\theta_M)\Big)_{l\in[1:L]}. \end{aligned} \quad (56)$$

To guarantee T_m-privacy, the index specified by the m^{th} user is protected by T_m random noise vectors. The queries sent from the m^{th} user to the n^{th} server are constructed as $Q_n^{(m,\theta_m)} = (Q_{n,1}^{(m,\theta_m)}, Q_{n,2}^{(m,\theta_m)}, \ldots, Q_{n,L}^{(m,\theta_m)})$ where

$$\begin{aligned} Q_{n,l}^{(m,\theta_m)} &= \mathbf{e}_{K_m}(\theta_m) + \sum_{t\in[1:T_m]} (f_l-\alpha_n)^t \mathbf{Z}_{m,t}^{(l)}, \\ &\forall l\in[1:L], m\in[1:M]. \end{aligned} \quad (57)$$

With the queries from the M users and stored $\mathbf{S}_n$, the n^{th} server first computes an intermediate result

$$\begin{aligned} &B_n^{(\theta_1,\theta_2,\ldots,\theta_M)} \\ &= \sum_{l\in[1:L]} \frac{1}{f_l-\alpha_n} \mathbf{S}_n^{(l)} \times_1 Q_{n,l}^{(1,\theta_1)} \times_2 Q_{n,l}^{(2\theta_2)} \times_3 \cdots \\ &\quad \cdots \times_M Q_{n,l}^{(M,\theta_M)} \\ &= \sum_{l\in[1:L]} \frac{1}{f_l-\alpha_n} \mathbf{W}^{(l)} \times_1 \mathbf{e}_{K_1}(\theta_1) \times_2 \cdots \times_M \mathbf{e}_{K_M}(\theta_M) \\ &\quad + I_0 + \alpha_n I_1 + \cdots + \alpha_n^{\sum T_m+X-1} I_{\sum T_m+X-1} \\ &= \sum_{l\in[1:L]} \frac{1}{f_l-\alpha_n} \mathbf{W}^{(l)}(\theta_1,\ldots,\theta_M) \\ &\quad + I_0 + \alpha_n I_1 + \cdots + \alpha_n^{\sum T_m+X-1} I_{\sum T_m+X-1}. \end{aligned} \quad (58)$$

As before, $I_0,\ldots,I_{\sum T_m+X-1}$ are $\sum T_m+X$ interference terms which are useless. Note that the distributive law applies here because mode-m multiplication is a multilinear operation. The highest order of α_n is $\sum T_m+X-1$, which results from

$$\begin{aligned} &\sum_{l\in[1:L]} \frac{1}{f_l-\alpha_n} (f_l-\alpha_n)^X \widehat{\mathbf{Z}}_{l,X} \times_1 (f_l-\alpha_n)^{T_1} \mathbf{Z}_{1,T_1}^{(l)} \\ &\qquad \times_2 \cdots \times_M (f_l-\alpha_n)^{T_M} \mathbf{Z}_{M,T_M}^{(l)}. \end{aligned} \quad (59)$$

Similar to DB-TPIR, the interference terms may contain some information of the indices specified by all users. To guarantee privacy between users, servers will add common randomness shared among them to the intermediate results to generate their answers for each user. Specifically, the answer from server n is

$$\begin{aligned} A_n^{(\theta_1,\ldots,\theta_M)} &= B_n^{(\theta_1,\ldots,\theta_M)} + \widetilde{Z}_0 + \alpha_n \widetilde{Z}_1 + \cdots \\ &\quad + \alpha_n^{\sum T_m+X-1} \widetilde{Z}_{\sum T_m+X-1} \end{aligned}$$

$$= \sum_{l\in[1:L]} \frac{1}{f_l-\alpha_n}\mathbf{W}^{(l)}(\theta_1,\ldots,\theta_M) + \underbrace{\left(I_0+\widetilde{Z}_0\right)}_{J_0} + \cdots + \alpha_n^{\sum T_m+X-1}\underbrace{\left(I_{\sum T_m+X-1}+\widetilde{Z}_{\sum T_m+X-1}\right)}_{J_{\sum T_m+X-1}}. \quad (60)$$

The matrix form of (60) is similar to (44), we omit it here. Since $L = N - \sum T_m - X$ dimensions are occupied by desired message symbols and $\sum T_m + X$ dimensions are occupied by the noisy versions of interference terms (J), the rate achieved here is

$$R = \frac{L}{N} = 1 - \frac{\sum T_m + X}{N}. \quad (61)$$

C. Example

Let us provide a simple example for illustration.

$N = 8$ *Servers,* $M = 3$ *users with* $T_1 = T_2 = 1, T_3 = 2$, $X = 2$, $L = 2$.

The storage at Server n and the queries from the 3 users are listed as follows.

	Server 'n'
$\mathbf{S}_n^{(1)}$	$\mathbf{W}^{(1)} + (f_1-\alpha_n)\widehat{\mathbf{Z}}_{1,1} + (f_1-\alpha_n)^2\widehat{\mathbf{Z}}_{1,2}$
$\mathbf{S}_n^{(2)}$	$\mathbf{W}^{(2)} + (f_2-\alpha_n)\widehat{\mathbf{Z}}_{2,1} + (f_2-\alpha_n)^2\widehat{\mathbf{Z}}_{2,2}$
$Q_{n,1}^{(1,\theta_1)}$	$\mathbf{e}_{K_1}(\theta_1) + (f_1-\alpha_n)\mathbf{Z}_{1,1}^{(1)}$
$Q_{n,2}^{(1,\theta_1)}$	$\mathbf{e}_{K_1}(\theta_1) + (f_2-\alpha_n)\mathbf{Z}_{1,1}^{(2)}$
$Q_{n,1}^{(2,\theta_2)}$	$\mathbf{e}_{K_2}(\theta_2) + (f_1-\alpha_n)\mathbf{Z}_{2,1}^{(1)}$
$Q_{n,2}^{(2,\theta_2)}$	$\mathbf{e}_{K_2}(\theta_2) + (f_2-\alpha_n)\mathbf{Z}_{2,1}^{(2)}$
$Q_{n,1}^{(3,\theta_3)}$	$\mathbf{e}_{K_3}(\theta_3) + (f_1-\alpha_n)\mathbf{Z}_{3,1}^{(1)} + (f_1-\alpha_n)^2\mathbf{Z}_{3,2}^{(1)}$
$Q_{n,2}^{(3,\theta_3)}$	$\mathbf{e}_{K_3}(\theta_3) + (f_2-\alpha_n)\mathbf{Z}_{3,1}^{(2)} + (f_2-\alpha_n)^2\mathbf{Z}_{3,2}^{(2)}$

The intermediate result is

$$B_n^{(\theta_1,\theta_2,\theta_3)} = \frac{1}{f_1-\alpha_n}\mathbf{W}^{(1)}(\theta_1,\theta_2,\theta_3) + \frac{1}{f_2-\alpha_n}\mathbf{W}^{(1)}(\theta_1,\theta_2,\theta_3) + I_0 + \cdots + \alpha_n^5 I_5.$$

The highest order of α is 5 since $T_1+T_2+T_3+X-1=5$ in this case. The answer from the server is

$$A_n^{(\theta_1,\theta_2,\theta_3)} = \frac{1}{f_1-\alpha_n}\mathbf{W}^{(1)}(\theta_1,\theta_2,\theta_3) + \frac{1}{f_2-\alpha_n}\mathbf{W}^{(2)}(\theta_1,\theta_2,\theta_3) + \underbrace{(I_0+\widetilde{Z}_0)}_{J_0} + \cdots + \alpha_n^5\underbrace{(I_5+\widetilde{Z}_5)}_{J_5}.$$

Evidently, the desired symbols occupy 2 dimensions, the aligned interference occupies 6 dimensions, and the rate achieved is $2/8 = 1/4$.

To further explain the example intuitively, $(\mathbf{S}_n^{(l)})_{l\in[1:2]}$ can be viewed as the secret shares of $\mathbf{W}^{(1)}, \mathbf{W}^{(2)}$ for the N servers, and $Q_{n,1}^{(1,\theta_1)}, Q_{n,2}^{(1,\theta_1)}$ can be viewed as two independent shares of $\mathbf{e}_{K_1}(\theta_1)$ at the n^{th} server, $n \in [1{:}N]$. Similarly, $Q_{n,1}^{(2,\theta_2)}, Q_{n,2}^{(2,\theta_2)}$ and $Q_{n,1}^{(3,\theta_3)}, Q_{n,2}^{(3,\theta_3)}$ are independent shares of $\mathbf{e}_{K_2}(\theta_2)$ and $\mathbf{e}_{K_3}(\theta_3)$, respectively. $B_n^{(\theta_1,\theta_2,\theta_3)}$ is constructed following the idea of CSA codes [61] such that the interference symbols align within the 6 dimensions of the subspace spanned by the Vandermonde terms, while the two desired symbols, represented as $\mathbf{W}^{(1)}(\theta_1,\theta_2,\theta_3) = \mathbf{W}^{(1)} \times_1 Q_{n,l}^{(1,\theta_1)} \times_2 Q_{n,l}^{(2\theta_2)} \times_3 Q_{n,l}^{(3,\theta_3)}$ and $\mathbf{W}^{(2)}(\theta_1,\theta_2,\theta_3) = \mathbf{W}^{(2)} \times_1 Q_{n,l}^{(1,\theta_1)} \times_2 Q_{n,l}^{(2\theta_2)} \times_3 Q_{n,l}^{(3,\theta_3)}$, remain resolvable along the Cauchy terms.

VI. Conclusion

We explored the problem of M-way blind X-secure T-private information retrieval (MB-XS-TPIR). We found the asymptotic capacity of double blind T-private information retrieval (DB-TPIR), which is a special case of MB-XS-TPIR, under a bounded-latency constraint. The achievable scheme was constructed based on Cross-Subspace Alignment. We then generalized the scheme using tensor-products into an MB-XS-TPIR scheme where the number of users (M), storage security-level (X) and privacy level of each user's index $(T_1, T_2, \ldots, T_M)$ can be arbitrarily chosen.

This work leads to a number of open problems. Foremost is the question of optimality of the proposed solutions. For example, the asymptotic capacity for MB-XS-TPIR remains open. For non-asymptotic settings, the capacity remains open even for DB-TPIR. As discussed in the introduction, we expect that our solution to MB-XS-TPIR may be asymptotically optimal. In fact, we expect that our solution may be optimal even in non-asymptotic settings. This is because of the constraint that the user must learn nothing about the other users' indices, which is reminiscent of 'symmetric' privacy constraints in PIR. Prior works, e.g., [12], [18], [72], [73], suggest that the capacity of PIR under symmetric privacy constraints tends to be the same as the asymptotic capacity without symmetric privacy constraints. Another open problem is to characterize the minimal amount of common randomness needed to be shared among servers for MB-XS-TPIR. Finally, yet another promising direction for future work is the setting of secure multiparty computation where the messages $\mathbf{W}(\theta_1,\theta_2,\ldots,\theta_M)$ are deterministic functions of $(\theta_1,\theta_2,\ldots,\theta_M)$. What makes these settings challenging is that their upload costs may not be negligible relative to download costs, so instead of a capacity figure the optimal solution may be a tradeoff between the upload and download costs.

Appendix

A. Proof of Corollary 1

Let us focus on (17), i.e., inter-user privacy from the 1^{st} user's perspective. Similar reasoning will apply to (18).

When $T_1 = T_2 = 1$, $N = L+2$, we neglect the t on superscripts or subscripts of all symbols. With this simplified notation, the private randomness of each of the two users can be expressed as

$$\mathcal{Z}_1 = \left\{\mathbf{Z}_1^{(l)} \mid l \in [1{:}L]\right\}, \mathcal{Z}_2 = \left\{\mathbf{Z}_2^{(l)} \mid l \in [1{:}L]\right\}.$$

The intermediate result computed by the n^{th} server can be written as

$$B_n^{(\theta_1,\theta_2)} = \frac{1}{f_1-\alpha_n}Q_{n,1}^{(1,\theta_1)\prime}\mathbf{W}^{(1)}Q_{n,1}^{(2,\theta_2)} + \cdots\cdots + \frac{1}{f_L-\alpha_n}Q_{n,L}^{(1,\theta_1)\prime}\mathbf{W}^{(L)}Q_{n,L}^{(2,\theta_2)} \quad (62)$$

$$= \frac{1}{f_1-\alpha_n}\mathbf{e}_{K_1}(\theta_1)'\mathbf{W}^{(1)}\mathbf{e}_{K_2}(\theta_2) + \cdots\cdots$$
$$+ \frac{1}{f_L-\alpha_n}\mathbf{e}_{K_1}(\theta_1)'\mathbf{W}^{(L)}\mathbf{e}_{K_2}(\theta_2) \tag{63}$$
$$+ \underbrace{\sum_{l\in[1:L]}\left(\mathbf{Z}_1^{(l)'}\mathbf{W}^{(l)}\mathbf{e}_{K_2}(\theta_2) + \mathbf{e}_{K_1}(\theta_1)'\mathbf{W}^{(l)}\mathbf{Z}_2^{(l)} + f_l\mathbf{Z}_1^{(l)'}\mathbf{W}^{(l)}\mathbf{Z}_2^{(l)}\right)}_{I_0} \tag{64}$$
$$+ \alpha_n \underbrace{\left(-\sum_{l\in[1:L]}\mathbf{Z}_1^{(l)'}\mathbf{W}^{(l)}\mathbf{Z}_2^{(l)}\right)}_{I_1}. \tag{65}$$

Note here that even though the expressions for I_0, I_1 are fairly involved, they are just 2 random variables in $\mathbb{F}_q$. Meanwhile, $B_{[1:N]}^{(\theta_1,\theta_2)}$ is an invertible function of $\mathbf{W}(\theta_1,\theta_2), I_0, I_1$.

Let us define three sets that contain all the components of I_0, I_1 except $\mathbf{e}_{K_1}(\theta_1)'\mathbf{W}^{(1)}\mathbf{Z}_2^{(1)}$ and $\mathbf{Z}_1^{(1)'}\mathbf{W}^{(1)}\mathbf{Z}_2^{(1)}$. Specifically,

$$\mathcal{I}_1 = \left\{\mathbf{Z}_1^{(l)'}\mathbf{W}^{(l)}\mathbf{e}_{K_2}(\theta_2) \mid l\in[1:L]\right\}, \tag{66}$$
$$\mathcal{I}_2 = \left\{\mathbf{e}_{K_1}(\theta_1)'\mathbf{W}^{(l)}\mathbf{Z}_2^{(l)} \mid l\in[2:L]\right\}, \tag{67}$$
$$\mathcal{I}_3 = \left\{\mathbf{Z}_1^{(l)'}\mathbf{W}^{(l)}\mathbf{Z}_2^{(l)} \mid l\in[2:L]\right\}. \tag{68}$$

So in q-ary units, we have

$$I\Big(\theta_2; B_{[1:N]}^{(\theta_1,\theta_2)} \mid \theta_1, \mathcal{Z}_1, \mathbf{W}(\theta_1,\theta_2)\Big) \tag{69}$$
$$= I(\theta_2; \mathbf{W}(\theta_1,\theta_2), I_0, I_1 \mid \theta_1, \mathcal{Z}_1, \mathbf{W}(\theta_1,\theta_2)) \tag{70}$$
$$= I(\theta_2; I_0, I_1 \mid \theta_1, \mathcal{Z}_1, \mathbf{W}(\theta_1,\theta_2)) \tag{71}$$
$$= H(I_0, I_1 \mid \theta_1, \mathcal{Z}_1, \mathbf{W}(\theta_1,\theta_2)) - H(I_0, I_1 \mid \theta_1, \mathcal{Z}_1, \mathbf{W}(\theta_1,\theta_2), \theta_2) \tag{72}$$
$$\le 2 - H\big(I_0, I_1 \mid \theta_1, \mathcal{Z}_1, \mathbf{W}(\theta_1,\theta_2), \theta_2, \mathcal{I}_{[1:3]}\big) \tag{73}$$
$$= 2 - H\Big(\mathbf{e}_{K_1}(\theta_1)'\mathbf{W}^{(1)}\mathbf{Z}_2^{(1)}, \mathbf{Z}_1^{(1)'}\mathbf{W}^{(1)}\mathbf{Z}_2^{(1)} \mid \theta_1, \mathcal{Z}_1, \mathbf{W}(\theta_1,\theta_2), \theta_2, \mathcal{I}_{[1:3]}\Big). \tag{74}$$

Equation (73) results from the fact that I_0, I_1 are in $\mathbb{F}_q$ and conditioning reduces entropy. Equation (74) holds because elements in $\mathcal{I}_{[1:3]}$ can be subtracted from I_0, I_1.

To proceed further we need to define the following new random variables.

$$E_1 = \begin{cases} 1, & \text{if } \mathbf{W}^{(1)} \text{ has full-rank,} \\ 0, & \text{otherwise.} \end{cases} \tag{75}$$
$$E_2 = \begin{cases} 1, & \text{if } \mathbf{Z}_1^{(1)} \neq \mathbf{0} \text{ and } \mathbf{Z}_1^{(1)} \perp\!\!\!\perp \mathbf{e}_{K_1}(\theta_1), \\ 0, & \text{otherwise.} \end{cases} \tag{76}$$

Recall that $\mathbf{Z}_1^{(1)} \perp\!\!\!\perp \mathbf{e}_{K_1}(\theta_1)$ denotes that the two vectors are linearly independent. We have

$$\Pr(E_1=1) = \frac{\prod_{i\in[1:K]}\left(q^K - q^{i-1}\right)}{q^{K^2}}, \tag{77}$$
$$\Pr(E_2=1) = 1 - \frac{1}{q^{K-1}}, \tag{78}$$
$$\Pr(E_1=1, E_2=1) = \Pr(E_1=1)\cdot\Pr(E_2=1). \tag{79}$$

Note that the numerator of (77) is the order of the general linear group of degree K over $\mathbb{F}_q$. Equation (79) follows because E_1 and E_2 are independent.

Consider the second term of (74), we have

$$H\Big(\mathbf{e}_{K_1}(\theta_1)'\mathbf{W}^{(1)}\mathbf{Z}_2^{(1)}, \mathbf{Z}_1^{(1)'}\mathbf{W}^{(1)}\mathbf{Z}_2^{(1)} \mid \theta_1, \mathcal{Z}_1, \mathbf{W}(\theta_1,\theta_2), \theta_2, \mathcal{I}_{[1:3]}\Big) \tag{80}$$
$$\ge H\Big(\mathbf{e}_{K_1}(\theta_1)'\mathbf{W}^{(1)}\mathbf{Z}_2^{(1)}, \mathbf{Z}_1^{(1)'}\mathbf{W}^{(1)}\mathbf{Z}_2^{(1)} \mid \theta_1, \mathcal{Z}_1, \mathbf{W}(\theta_1,\theta_2), \theta_2, \mathcal{I}_{[1:3]}, E_1, E_2\Big) \tag{81}$$
$$\ge \Pr(E_1=1, E_2=1)\, H\Big(\mathbf{e}_{K_1}(\theta_1)'\mathbf{W}^{(1)}\mathbf{Z}_2^{(1)}, \mathbf{Z}_1^{(1)'}\mathbf{W}^{(1)}\mathbf{Z}_2^{(1)} \mid \theta_1, \mathcal{Z}_1, \mathbf{W}(\theta_1,\theta_2), \theta_2, \mathcal{I}_{[1:3]}, E_1=1, E_2=1\Big). \tag{82}$$

Let $\mathbf{R}_1, \mathbf{R}_2$ be two row vectors and

$$\mathbf{R}_1 = \mathbf{e}_{K_1}(\theta_1)'\mathbf{W}^{(1)}, \mathbf{R}_2 = \mathbf{Z}_1^{(1)'}\mathbf{W}^{(1)}. \tag{83}$$

$E_1 = 1$ implies that $\mathbf{W}^{(1)}$ has full-rank. $E_2 = 1$ means that $\mathbf{Z}_1^{(1)}$ and $\mathbf{e}_{K_1}(\theta_1)$ are linearly independent. So $\mathbf{R}_1, \mathbf{R}_2$ are linearly independent. Let $(i,j) \in [1:K]\times[1:K], i\neq j$ be the smallest pair such that

$$\mathbf{M} = \begin{bmatrix} \mathbf{R}_1(i) & \mathbf{R}_1(j) \\ \mathbf{R}_2(i) & \mathbf{R}_2(j) \end{bmatrix}, \det(\mathbf{M}) \neq 0. \tag{84}$$

Such (i,j) must exist due to the linear independence of $\mathbf{R}_1$ and $\mathbf{R}_2$.

Let $\overline{\mathcal{Z}_2} = \{\mathbf{Z}_2^{(1)}(k) \mid k\in[1:K]\backslash\{i,j\}\}$ contain all the entries of $\mathbf{Z}_2^{(1)}$ except $\mathbf{Z}_2^{(1)}(i), \mathbf{Z}_2^{(1)}(j)$, for (82), we have

$$2 \ge H\Big(\mathbf{e}_{K_1}(\theta_1)'\mathbf{W}^{(1)}\mathbf{Z}_2^{(1)}, \mathbf{Z}_1^{(1)'}\mathbf{W}^{(1)}\mathbf{Z}_2^{(1)} \mid \theta_1, \mathcal{Z}_1, \mathbf{W}(\theta_1,\theta_2), \theta_2, \mathcal{I}_{[1:3]}, E_1=1, E_2=1\Big) \tag{85}$$
$$= H\Big(\mathbf{R}_1\mathbf{Z}_2^{(1)}, \mathbf{R}_2\mathbf{Z}_2^{(1)} \mid \theta_1, \mathcal{Z}_1, \mathbf{W}(\theta_1,\theta_2), \theta_2, \mathcal{I}_{[1:3]}, E_1=1, E_2=1\Big) \tag{86}$$
$$\ge H\Big(\mathbf{R}_1\mathbf{Z}_2^{(1)}, \mathbf{R}_2\mathbf{Z}_2^{(1)} \mid \theta_1, \mathcal{Z}_1, \mathbf{W}(\theta_1,\theta_2), \theta_2, \mathcal{I}_{[1:3]}, E_1=1, E_2=1, \mathbf{R}_1, \mathbf{R}_2, i, j, \overline{\mathcal{Z}_2}\Big) \tag{87}$$
$$= H\left(\mathbf{M}.\begin{bmatrix}\mathbf{Z}_2^{(1)}(i)\\ \mathbf{Z}_2^{(1)}(j)\end{bmatrix} \Big| \theta_1, \mathcal{Z}_1, \mathbf{W}(\theta_1,\theta_2), \theta_2, \mathcal{I}_{[1:3]}, E_1=1, E_2=1, \mathbf{R}_1, \mathbf{R}_2, i, j, \overline{\mathcal{Z}_2}\right) \tag{88}$$
$$= H\Big(\mathbf{Z}_2^{(1)}(i), \mathbf{Z}_2^{(1)}(j) \mid \theta_1, \mathcal{Z}_1, \mathbf{W}(\theta_1,\theta_2), \theta_2, \mathcal{I}_{[1:3]}, E_1=1, E_2=1, \mathbf{R}_1, \mathbf{R}_2, i, j, \overline{\mathcal{Z}_2}\Big) = 2 \tag{89}$$

in q-ary units. Equation (88) holds because we can subtract other components of $\mathbf{R}_1\mathbf{Z}_2^{(1)}, \mathbf{R}_2\mathbf{Z}_2^{(1)}$ given the conditioned terms. Equation (89) results from the fact that $\mathbf{M}$ is invertible and $\mathbf{Z}_2^{(1)}(i)$, $\mathbf{Z}_2^{(1)}(j)$ are independent of all conditioned terms.

So for the second term of (74) we have

$$H\Big(\mathbf{e}_{K_1}(\theta_1)'\mathbf{W}^{(1)}\mathbf{Z}_2^{(1)}, \mathbf{Z}_1^{(1)'}\mathbf{W}^{(1)}\mathbf{Z}_2^{(1)} \mid \theta_1, \mathcal{Z}_1, \mathbf{W}(\theta_1,\theta_2), \theta_2, \mathcal{I}_{[1:3]}\Big) \ge 2\Pr(E_1=1, E_2=1). \tag{90}$$

Combining (90) with (74), we have

$$I\Big(\theta_2; B_{[1:N]}^{(\theta_1,\theta_2)} \mid \theta_1, \mathcal{Z}_1, \mathbf{W}(\theta_1,\theta_2)\Big) \tag{91}$$

$$\leq 2(1-\Pr(E_1=1,E_2=1)) \tag{92}$$

$$= 2\left(1-\left(1-\frac{1}{q^{K-1}}\right)\frac{\prod_{k\in[1:K]}\left(q^K-q^{k-1}\right)}{q^{K^2}}\right) \tag{93}$$

$$\leq 2\left(1-\left(1-\frac{1}{q^{K-1}}\right)\frac{\left(q^K-q^{K-1}\right)^K}{q^{K^2}}\right) \tag{94}$$

$$= 2\left(1-\left(1-\frac{1}{q^{K-1}}\right)\left(1-\frac{1}{q}\right)^K\right). \tag{95}$$

To ensure that the LHS of (91) is bounded above by ϵ for $q > q_0$, we can choose q_0 to be any value of q that bounds the RHS of (95) above by ϵ. ■

B. Proof of Corollary 2

The lower-bound follows already from the proof of achievability of Theorem 2. Here we prove the upper bound. Any MB-XS-TPIR scheme with parameters $K_1,\ldots,K_M,T_1,\ldots,T_M$ yields a total of M XS-TPIR schemes. For the m^{th} XS-TPIR scheme where $m\in[1{:}M]$, the user corresponds to the m^{th} user of MB-XS-TPIR. All other users in MB-XS-TPIR generate fixed indices so that the user is retrieving a message in a database with K_m messages, i.e., $\mathbf{W}(i_1,\ldots,i_{m-1},\theta_m,i_{m+1},\ldots,i_M)$ where $\theta_m\in[1{:}K_m]$ while $i_1,\ldots,i_{m-1},i_{m+1},\ldots,i_M$ are fixed, subject to T_m-privacy constraint from N servers. The rate of MB-XS-TPIR cannot exceed $\frac{1-\frac{T_m+X}{N}}{1-(\frac{T_m}{N-X})^{K_m}}$ because this value is the upper bound of the achievable rates of XS-TPIR with N servers, K_m messages and T_m-privacy constraint according to [29]. Since this upper bound holds for all $m\in[1{:}M]$, the upper bound of (20) follows. ■

REFERENCES

[1] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[2] Y. Gertner, S. Goldwasser, and T. Malkin, "A random server model for private information retrieval," in *Randomization and Approximation Techniques in Computer Science*. Heidelberg, Germany: Springer, 1998, pp. 200–217.

[3] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," in *Proc. 36th Annu. Symp. Found. Comput. Sci.*, 1995, pp. 41–50.

[4] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," *J. ACM*, vol. 45, no. 6, pp. 965–981, 1998.

[5] A. C. Yao, "Protocols for secure computations," in *Proc. IEEE 23rd Annu. Symp. Found. Comput. Sci. (SFCS)*, 1982, pp. 160–164.

[6] A. C.-C. Yao, "How to generate and exchange secrets," in *Proc. IEEE 27th Annu. Symp. Found. Comput. Sci. (SFCS)*, 1986, pp. 162–167.

[7] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game, or a completeness theorem for protocols with honest majority," in *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*. New York, NY, USA: Assoc. Comput. Mach., 2019, pp. 307–328.

[8] U. Feige, J. Killian, and M. Naor, "A minimal model for secure computation," in *Proc. 26th Annu. ACM Symp. Theory Comput.*, 1994, pp. 554–563.

[9] T. H. Chan, S.-W. Ho, and H. Yamamoto, "Private information retrieval for coded storage," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2015, pp. 2842–2846.

[10] H. Sun and S. A. Jafar, "The capacity of private information retrieval," *IEEE Trans. Inf. Theory*, vol. 63, no. 7, pp. 4075–4088, Jul. 2017.

[11] H. Sun and S. A. Jafar, "The capacity of robust private information retrieval with colluding databases," *IEEE Trans. Inf. Theory*, vol. 64, no. 4, pp. 2361–2370, Apr. 2018.

[12] H. Sun and S. A. Jafar, "The capacity of symmetric private information retrieval," *IEEE Trans. Inf. Theory*, vol. 65, no. 1, pp. 322–329, Jan. 2019.

[13] I. Samy, M. A. Attia, R. Tandon, and L. Lazos, "Asymmetric leaky private information retrieval," 2020. [Online]. Available: arXiv:2006.03048.

[14] H.-Y. Lin *et al.*, "Multi-server weakly-private information retrieval," 2020. [Online]. Available: arXiv:2007.10174.

[15] Q. Wang and M. Skoglund, "Secure private information retrieval from colluding databases with eavesdroppers," 2017. [Online]. Available: arXiv:1710.01190.

[16] Q. Wang, H. Sun, and M. Skoglund, "The capacity of private information retrieval with eavesdroppers," *IEEE Trans. Inf. Theory*, vol. 65, no. 5, pp. 3198–3214, May 2019.

[17] Q. Wang and M. Skoglund, "On PIR and symmetric PIR from colluding databases with adversaries and eavesdroppers," *IEEE Trans. Inf. Theory*, vol. 65, no. 5, pp. 3183–3197, May 2019.

[18] Q. Wang, H. Sun, and M. Skoglund, "The ϵ-error capacity of symmetric PIR with Byzantine adversaries," 2018. [Online]. Available: arXiv:1809.03988.

[19] K. Banawan and S. Ulukus, "The capacity of private information retrieval from Byzantine and colluding databases," *IEEE Trans. Inf. Theory*, vol. 65, no. 2, pp. 1206–1219, Feb. 2019.

[20] R. Freij-Hollanti, O. Gnilke, C. Hollanti, and D. Karpuk, "Private information retrieval from coded databases with colluding servers," *SIAM J. Appl. Algebra Geometry*, vol. 1, no. 1, pp. 647–664, 2017.

[21] R. Tajeddine, O. W. Gnilke, D. Karpuk, R. Freij-Hollanti, C. Hollanti, and S. E. Rouayheb, "Private information retrieval schemes for coded data with arbitrary collusion patterns," 2017. [Online]. Available: arXiv:1701.07636.

[22] Q. Wang and M. Skoglund, "Linear symmetric private information retrieval for MDS coded distributed storage with colluding servers," 2017. [Online]. Available: arXiv:1708.05673.

[23] 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 Trans. Inf. Theory*, vol. 64, no. 2, pp. 1000–1022, Feb. 2018.

[24] R. Tajeddine, O. W. Gnilke, D. Karpuk, R. Freij-Hollanti, and C. Hollanti, "Private information retrieval from coded storage systems with colluding, Byzantine, and unresponsive servers," *IEEE Trans. Inf. Theory*, vol. 65, no. 6, pp. 3898–3906, Jun. 2019.

[25] Q. Wang and M. Skoglund, "Symmetric private information retrieval from MDS coded distributed storage with non-colluding and colluding servers," *IEEE Trans. Inf. Theory*, vol. 65, no. 8, pp. 5160–5175, Aug. 2019.

[26] Z. Jia and S. A. Jafar, "X-secure T-private information retrieval from MDS coded storage with Byzantine and unresponsive servers," *IEEE Trans. Inf. Theory*, vol. 66, no. 12, pp. 7427–7438, Dec. 2020.

[27] R. Zhou, C. Tian, H. Sun, and T. Liu, "Capacity-achieving private information retrieval codes from MDS-coded databases with minimum message size," *IEEE Trans. Inf. Theory*, vol. 66, no. 8, pp. 4904–4916, Aug. 2020.

[28] H. Yang, W. Shin, and J. Lee, "Private information retrieval for secure distributed storage systems," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 12, pp. 2953–2964, Dec. 2018.

[29] Z. Jia, H. Sun, and S. A. Jafar, "Cross subspace alignment and the asymptotic capacity of X-secure T-private information retrieval," *IEEE Trans. Inf. Theory*, vol. 65, no. 9, pp. 5783–5798, Sep. 2019.

[30] Z. Jia and S. A. Jafar, "On the asymptotic capacity of X-secure T-private information retrieval with graph based replicated storage," *IEEE Trans. Inf. Theory*, vol. 66, no. 10, pp. 6280–6296, Oct. 2020.

[31] R. T. Mohamed Adel Attia, Deepak Kumar, "The capacity of private information retrieval from uncoded storage constrained databases," 2018. [Online]. Available: arXiv:1805.04104.

[32] Y.-P. Wei, B. Arasli, K. Banawan, and S. Ulukus, "The capacity of private information retrieval from decentralized uncoded caching databases," *Information*, vol. 10, no. 12, p. 372, 2019.

[33] N. Woolsey, R.-R. Chen, and M. Ji, "Private information retrieval from heterogeneous uncoded storage constrained databases with reduced submessages," 2019. [Online]. Available: arXiv:1904.02131.

[34] T. Guo, R. Zhou, and C. Tian, "New results on the storage-retrieval trade-off in private information retrieval systems," 2020. [Online]. Available: arXiv:2008.00960.

[35] K. Banawan, B. Arasli, Y.-P. Wei, and S. Ulukus, "The capacity of private information retrieval from heterogeneous uncoded caching databases," *IEEE Trans. Inf. Theory*, vol. 66, no. 6, pp. 3407–3416, Jun. 2020.
[36] R. Tandon, "The capacity of cache aided private information retrieval," 2017. [Online]. Available: arXiv:1706.07035.
[37] Y.-P. Wei, K. Banawan, and S. Ulukus, "The capacity of private information retrieval with partially known private side information," *IEEE Trans. Inf. Theory*, vol. 65, no. 12, pp. 8222–8231, Dec. 2019.
[38] Y. Wei, K. Banawan, and S. Ulukus, "Fundamental limits of cache-aided private information retrieval with unknown and uncoded prefetching," *IEEE Trans. Inf. Theory*, vol. 65, no. 5, pp. 3215–3232, May 2019.
[39] Z. Chen, Z. Wang, and S. Jafar, "The capacity of T-private information retrieval with private side information," *IEEE Trans. Inf. Theory*, vol. 66, no. 8, pp. 4761–4773, Aug. 2020.
[40] H. Sun and S. A. Jafar, "Multiround private information retrieval: Capacity and storage overhead," *IEEE Trans. Inf. Theory*, vol. 64, no. 8, pp. 5743–5754, Aug. 2018.
[41] X. Yao, N. Liu, and W. Kang, "The capacity of multi-round private information retrieval from Byzantine databases," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2019, pp. 2124–2128.
[42] K. Banawan and S. Ulukus, "Multi-message private information retrieval: Capacity results and near-optimal schemes," *IEEE Trans. Inf. Theory*, vol. 64, no. 10, pp. 6842–6862, Oct. 2018.
[43] S. P. Shariatpanahi, M. J. Siavoshani, and M. A. Maddah-Ali, "Multi-message private information retrieval with private side information," 2018. [Online]. Available: arXiv:1805.11892.
[44] Z. Jia and S. Jafar, "On the capacity of secure distributed matrix multiplication," 2019. [Online]. Available: arXiv:1908.06957.
[45] Z. Wang, K. Banawan, and S. Ulukus, "Private set intersection: A multi-message symmetric private information retrieval perspective," 2019. [Online]. Available: arXiv:1912.13501.
[46] C. Tian, H. Sun, and J. Chen, "Capacity-achieving private information retrieval codes with optimal message size and upload cost," *IEEE Trans. Inf. Theory*, vol. 65, no. 11, pp. 7613–7627, Nov. 2019.
[47] X. Yao, N. Liu, and W. Kang, "The capacity of private information retrieval under arbitrary collusion patterns," 2020. [Online]. Available: arXiv:2001.03843.
[48] S. Li and M. Gastpar, "Single-server multi-message private information retrieval with side information," 2018. [Online]. Available: arXiv:1808.05797.
[49] S. Li and M. Gastpar, "Single-server multi-user private information retrieval with side information," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2018, pp. 1954–1958.
[50] S. Kadhe, A. Heidarzadeh, A. Sprintson, and O. O. Koyluoglu, "On an equivalence between single-server PIR with side information and locally recoverable codes," 2019. [Online]. Available: arXiv:1907.00598.
[51] A. Heidarzadeh, F. Kazemi, and A. Sprintson, "Capacity of single-server single-message private information retrieval with coded side information," in *Proc. ITW*, 2018, pp. 1–5.
[52] A. Heidarzadeh, F. Kazemi, and A. Sprintson, "Capacity of single-server single-message private information retrieval with private coded side information," in *Proc. ISIT*, 2019, pp. 1662–1666.
[53] F. Kazemi, E. Karimi, A. Heidarzadeh, and A. Sprintson, "Single-server single-message online private information retrieval with side information," 2019. [Online]. Available: arXiv:1901.07748.
[54] A. Heidarzadeh, S. Kadhe, S. El Rouayheb, and A. Sprintson, "Single-server multi-message individually-private information retrieval with side information," 2019. [Online]. Available: arXiv:1901.07509.
[55] I. Samy, M. A. Attia, R. Tandon, and L. Lazos, "Latent-variable private information retrieval," 2020. [Online]. Available: arXiv:2001.05998.
[56] M. Mirmohseni and M. A. Maddah-Ali, "Private function retrieval," 2017. [Online]. Available: arXiv:1711.04677.
[57] H. Sun and S. A. Jafar, "The capacity of private computation," *IEEE Trans. Inf. Theory*, vol. 65, no. 6, pp. 3880–3897, Jun. 2019.
[58] M. H. Mousavi, M. A. Maddah-Ali, and M. Mirmohseni, "Private inner product retrieval for distributed machine learning," 2019. [Online]. Available: arXiv:1902.06319.
[59] S. A. Obead, H.-Y. Lin, E. Rosnes, and J. Kliewer, "Private function computation for noncolluding coded databases," 2020. [Online]. Available: arXiv:2003.10007.
[60] Z. Chen, Z. Wang, and S. A. Jafar, "The asymptotic capacity of private search," *IEEE Trans. Inf. Theory*, vol. 66, no. 8, pp. 4709–4721, Aug. 2020.
[61] Z. Jia and S. Jafar, "Cross-subspace alignment codes for coded distributed batch computation," 2019. [Online]. Available: arXiv:1909.13873.
[62] H. Sun and S. A. Jafar, "On the capacity of locally decodable codes," *IEEE Trans. Inf. Theory*, vol. 66, no. 10, pp. 6566–6579, Oct. 2020.
[63] Y. Ishai and E. Kushilevitz, "On the hardness of information-theoretic multiparty computation," in *Proc. Adv. Cryptol. EUROCRYPT*, 2004, pp. 439–455.
[64] J. Kakar, S. Ebadifar, and A. Sezgin, "On the capacity and straggler-robustness of distributed secure matrix multiplication," *IEEE Access*, vol. 7, pp. 45783–45799, 2019.
[65] Z. Chen, Z. Jia, Z. Wang, and S. A. Jafar, "GCSA codes with noise alignment for secure coded multi-party batch matrix multiplication," 2020. [Online]. Available: arXiv:2002.07750.
[66] V. Cadambe and P. Grover, "Codes for distributed computing: A tutorial," *IEEE ITSOC Newslett.*, vol. 67, no. 4, pp. 3–15, Dec. 2017.
[67] Q. Yu, S. Li, N. Raviv, S. M. M. Kalan, M. Soltanolkotabi, and S. Avestimehr, "Lagrange coded computing: Optimal design for resiliency, security and privacy," 2018. [Online]. Available: arXiv:1806.00939.
[68] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Polynomial codes: An optimal design for high-dimensional coded matrix multiplication," 2017. [Online]. Available: arXiv:1705.10464.
[69] E. D. Habil, "Double sequences and double series," *IUG J. Nat. Stud.*, vol. 14, no. 1, p. 32, 2016.
[70] M. Gasca, J. Martinez, and G. Mühlbach, "Computation of rational interpolants with prescribed poles," *J. Comput. Appl. Math.*, vol. 26, no. 3, pp. 297–309, 1989.
[71] H. Lu, K. N. Plataniotis, and A. Venetsanopoulos, *Multilinear Subspace Learning: Dimensionality Reduction of Multidimensional Data*. New York, NY, USA: CRC Press, 2013.
[72] Q. Wang and M. Skoglund, "Symmetric private information retrieval for MDS coded distributed storage," 2016. [Online]. Available: arXiv:1610.04530.
[73] Q. Wang and M. Skoglund, "Secure symmetric private information retrieval from colluding databases with adversaries," 2017. [Online]. Available: arXiv:1707.02152.