Private Linear Transformation: The Individual Privacy Case

Nahid Esmati, Anoosheh Heidarzadeh, and Alex Sprintson

***Abstract*— This paper considers the single-server Private Linear Transformation (PLT) problem when individual privacy is required. In this problem, there is a user that wishes to obtain L linear combinations of a D-subset of messages belonging to a dataset of K messages stored on a single server. The goal is to minimize the download cost while keeping the identity of every message required for the computation individually private. We focus on the setting in which the matrix of coefficients pertaining to the required linear combinations is the generator matrix of a maximum distance separable code. We establish lower and upper bounds on the capacity of PLT with individual privacy, where the capacity is defined as the supremum of all achievable download rates. We show that our bounds are tight under certain divisibility conditions. In addition, we present lower bounds on the capacity of the settings in which the user has a prior side information about a subset of messages.**

2021 IEEE International Symposium on Information Theory (ISIT) | 978-1-5386-8209-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISIT45174.2021.9517762

I. INTRODUCTION

In this work, we study the problem of single-server *Private Linear Transformation (PLT) with Individual Privacy*, referred to as *IPLT* for short. In this problem, there is a single server that stores a set of K messages, and a user that wants to compute L linear combinations of a subset of D messages. This setup appears in several practical scenarios such as linear transformation for dimensionality reduction in Machine Learning (ML) applications (for a detailed example, see a long version of this work, [1]). The objective of the user is to recover the required linear combinations by downloading minimum possible amount of information from the server, while protecting the privacy of the identity of every message required for the computation individually. The individual privacy requirement implies that, from the perspective of the server, every message is equally likely to belong to the D-subset of messages that constitute the support set of the required linear combinations.

The IPLT problem is related to the single-server PLT with Joint Privacy (JPLT) problem, which we have studied in a parallel work [2]. The joint privacy condition implies that, from the server's perspective, any D-subset of messages must be equally likely to be the support set of the required linear combinations. Joint privacy was previously considered in [3]–[6] for Private Information Retrieval (PIR), and in [7], [8] for Private Linear Computation (PLC). Individual privacy is a relaxed version of joint privacy with an operational meaning, and is motivated by the need to protect the access pattern for individual messages. This is of practical importance in many scenarios. For example, consider a setting in which the dataset contains information about individuals, and the user is required to hide information on whether the data belonging to an individual was used in the computation. Note that both of these privacy conditions are weaker than the privacy condition considered in [9]–[12] for multi-server PLC, where the values of the combination coefficients in the required linear combination must be kept private. That said, joint and individual privacy may still provide a satisfactory degree of privacy in many practical scenarios.

Individual privacy was originally introduced in [13] for PIR with Individual Privacy (IPIR), and was recently considered for PLC with Individual Privacy (IPLC) in [8]. Note that IPLT reduces to IPIR or IPLC for $L = D$ or $L = 1$, respectively. IPIR and IPLC were studied in the settings in which the user has a prior side information about a subset of messages. It was shown that, when compared to PIR and PLC with joint privacy, IPIR and IPLC can be performed with a much lower download cost. Motivated by these results, this work seeks to answer the following questions: (i) can IPLT be performed with a lower download cost than JPLT? (ii) can a prior side information be leveraged to further decrease the download cost of IPLT? (iii) what are the fundamental limits on the download cost for IPLT? We make a significant progress towards answering these questions in this work.

A. Main Contributions

In this work, we focus on the setting in which the coefficient matrix corresponding to the required linear combinations is the generator matrix of a Maximum Distance Separable (MDS) code. The MDS coefficient matrices are motivated by the application of *random linear transformation* for dimensionality reduction in ML, see, e.g., [14]. When the operations are performed over the field of reals (or a sufficiently large finite field), a random transformation matrix is MDS with probability 1 (or with high probability). For this setting, we establish lower and upper bounds on the capacity of IPLT, where the capacity is defined as the supremum of all achievable download rates. In addition, we show that our bounds are tight under certain divisibility conditions, settling the capacity of IPLT for such cases.

To prove the upper bound on the capacity, we use information-theoretic arguments based on a necessary condition for IPLT schemes, and formulate the problem as an integer linear programming (ILP) problem. Solving this ILP, we obtain the capacity upper bound. The lower bound on the capacity is proven by a novel achievability scheme, termed *Generalized Partition-and-Code with Partial Interference Alignment (GPC-PIA) protocol*. This protocol generalizes the protocols we previously proposed in [13] and [8] for IPIR and IPLC, respectively. In addition, we present lower bounds

The authors are with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA (E-mail: {nahid, anoosheh, spalex}@tamu.edu).

on the capacity of the settings in which the user has a prior side information about a subset of messages. Our results indicate that, when there is no side information, IPLT can be performed more efficiently than JPLT, in terms of the download cost. The advantage of IPLT over JPLT is even more pronounced when the user knows a subset of messages or a subspace spanned by them as side information.

II. Problem Setup

Throughout, we denote random variables and their realizations by bold-face and regular symbols, respectively.

Let $\mathbb{F}_p$ be a finite field of order p, and let $\mathbb{F}_p^\ell$ be an ℓ-dimensional vector space over $\mathbb{F}_p$. Let K, D, L be positive integers such that $L \leq D \leq K$, and let $\mathcal{K} \triangleq \{1, ..., K\}$. We denote by $\mathcal{W}$ the set of all D-subsets of $\mathcal{K}$, and denote by $\mathcal{V}$ the set of all $L \times D$ matrices (with entries from $\mathbb{F}_p$) each of which generates an MDS code.

Suppose that there is a server that stores K messages $X_1, \ldots, X_K$, where $X_i \in \mathbb{F}_p^\ell$ for $i \in \mathcal{K}$ is a row-vector of length ℓ. Let $\mathrm{X} \triangleq [X_1^\mathsf{T}, \ldots, X_K^\mathsf{T}]^\mathsf{T}$. For every $\mathrm{S} \subset \mathcal{K}$, we denote by X_S the matrix X restricted to its rows indexed by S. We assume that $\mathbf{X}_1, \ldots, \mathbf{X}_K$ are independently and uniformly distributed over $\mathbb{F}_p^\ell$. That is, $H(\mathbf{X}_i) = \theta \triangleq \ell \log_2 p$ for $i \in \mathcal{K}$, and $H(\mathbf{X}_\mathrm{S}) = |\mathrm{S}|\theta$ for $\mathrm{S} \subset \mathcal{K}$, where $|\mathrm{S}|$ denotes the size of S. Note that $H(\mathbf{X}) = K\theta$. Suppose that there is a user that wants to compute the matrix $\mathrm{Z}^{[\mathrm{W},\mathrm{V}]} \triangleq \mathrm{V}\mathrm{X}_\mathrm{W}$, where $\mathrm{W} \in \mathcal{W}$ and $\mathrm{V} \in \mathcal{V}$. That is, $\mathrm{Z}^{[\mathrm{W},\mathrm{V}]}$ contains L rows $\mathrm{v}_1\mathrm{X}_\mathrm{W}, \ldots, \mathrm{v}_L\mathrm{X}_\mathrm{W}$, where v_l is the lth row of the matrix V. Note that $H(\mathbf{Z}^{[\mathrm{W},\mathrm{V}]}) = L\theta$. We refer to $\mathrm{Z}^{[\mathrm{W},\mathrm{V}]}$ as the *demand*, W as the *support index set of the demand*, V as the *coefficient matrix of the demand*, D as the *support size of the demand*, and L as the *dimension of the demand*.

We assume that (i) $\mathbf{W}, \mathbf{V}, \mathbf{X}$ are independent; (ii) $\mathbf{W}$ and $\mathbf{V}$ are uniformly distributed over $\mathcal{W}$ and $\mathcal{V}$, respectively; and (iii) the parameters D and L, and the distribution of $(\mathbf{W}, \mathbf{V})$ are initially known by the server, whereas the realization (W, V) is not initially known by the server.

Given (W, V), the user generates a query $\mathrm{Q} = \mathrm{Q}^{[\mathrm{W},\mathrm{V}]}$, which is a (potentially stochastic) function of (W, V), and sends it to the server. The query Q must satisfy the following privacy condition: given the query Q, every message index must be equally likely to belong to the demand's support index set. That is, for every $i \in \mathcal{K}$, it must hold that

$$\Pr(i \in \mathbf{W} | \mathbf{Q} = \mathrm{Q}) = \Pr(i \in \mathbf{W}),$$

where $\mathbf{Q}$ denotes $\mathbf{Q}^{[\mathbf{W},\mathbf{V}]}$. This condition—which was recently introduced in [13] and [8] for single-server PIR and PLC, is referred to as the *individual privacy condition*.

Upon receiving the query Q, the server generates an answer $\mathrm{A} = \mathrm{A}^{[\mathrm{W},\mathrm{V}]}$, and sends it back to the user. The answer A is a deterministic function of Q and X. The collection of the answer A, the query Q, and the realization (W, V) must enable the user to recover $\mathrm{Z}^{[\mathrm{W},\mathrm{V}]}$. That is,

$$H(\mathbf{Z} | \mathbf{A}, \mathbf{Q}, \mathbf{W}, \mathbf{V}) = 0,$$

where $\mathbf{Z}$ and $\mathbf{A}$ denote $\mathbf{Z}^{[\mathbf{W},\mathbf{V}]}$ and $\mathbf{A}^{[\mathbf{W},\mathbf{V}]}$, respectively. This condition is referred to as the *recoverability condition*.

We would like to design a protocol for generating a query $\mathrm{Q}^{[\mathrm{W},\mathrm{V}]}$ and the corresponding answer $\mathrm{A}^{[\mathrm{W},\mathrm{V}]}$ such that the individual privacy and recoverability conditions are satisfied. We refer to this problem as single-server *Private Linear Transformation (PLT) with Individual Privacy*, or *IPLT* for short. We define the *rate* of an IPLT protocol as the ratio of the entropy of the demand (i.e., $H(\mathbf{Z}) = L\theta$) to the entropy of the answer (i.e., $H(\mathbf{A})$). We define the *capacity* of the IPLT setting as the supremum of rates over all IPLT protocols. In this work, our goal is to establish (preferably matching) lower and upper bounds (in terms of K, D, L) on the capacity of the IPLT setting.

III. A Necessary Condition for IPLT Protocols

The individual privacy and recoverability conditions yield a necessary (but not sufficient) condition for any IPLT protocol, stated in Lemma 1. The proof is straightforward by the way of contradiction, and hence, omitted.

Lemma 1. *Given any IPLT protocol, for any $i \in \mathcal{K}$, there must exist $\mathrm{W}^* \in \mathcal{W}$ with $i \in \mathrm{W}^*$, and $\mathrm{V}^* \in \mathcal{V}$, such that*

$$H(\mathbf{Z}^{[\mathrm{W}^*,\mathrm{V}^*]} | \mathbf{A}, \mathbf{Q}) = 0.$$

The result of Lemma 1 establishes a connection between *linear codes* with a certain constraint and *linear schemes* for IPLT, i.e., any scheme in which the server's answer to the user's query consists of only *linear* combinations of the messages. In particular, the matrix of combination coefficients—pertaining to the linear combinations in the answer, must be the generator matrix of a linear code of length K that satisfies the following condition: for any coordinate i, there must exist $K - D$ coordinates different from i such that the code resulting from puncturing[1] at these $K - D$ coordinates contains L codewords that are MDS. Note, however, that this condition is only necessary and not sufficient. In particular, a sufficient (yet not necessary) condition is that, for *every coordinate* i, the punctured codes resulting from puncturing at any $K-D$ other coordinates (different from i) collectively contain the *same number of groups* of L codewords that are MDS. Maximizing the rate of a linear IPLT scheme is then equivalent to minimizing the dimension of a linear code that satisfies this sufficient condition. However, despite the fact that this sufficient condition is stronger than the necessary condition provided by Lemma 1, the former is more combinatorial, while the latter is more information-theoretic and hence more useful in the converse proof.

IV. Main Results

This section summarizes our main results for IPLT.

Theorem 1. *For the IPLT setting with K messages, demand's support size D, and demand's dimension L, the capacity is lower and upper bounded by $(\lfloor \frac{K}{D} \rfloor + \min\{\frac{R}{S}, \frac{R}{L}\})^{-1}$ and $(\lfloor \frac{K}{D} \rfloor + \min\{1, \frac{R}{L}\})^{-1}$, respectively, where $R \triangleq K \pmod D$ and $S \triangleq \gcd(D + R, R)$. The lower and upper bounds match when $R \leq L$ or $R \mid D$.*

[1]To puncture a linear code at a coordinate, the column corresponding to that coordinate is deleted from the generator matrix of the code.

To prove the converse bound, we use the necessary condition for IPLT protocols provided by Lemma 1 along with information-theoretic arguments, and formulate the problem as an integer linear programming (ILP) problem. Solving this ILP, we obtain the upper bound on the capacity (see Section V). The lower bound on the capacity is proven by constructing an IPLT protocol, called *Generalized Partition-and-Code with Partial Interference Alignment (GPC-PIA)* (see Section VI). This protocol is a generalization of the protocols we previously proposed in [13] and [8] for single-server PIR and PLC (without SI) with individual privacy. The main ingredients of the GPC-PIA protocol are as follows: (i) constructing a properly designed family of subsets of messages, where some subsets are possibly overlapping, and (ii) designing a number of linear combinations for each subset, where the linear combinations pertaining to the overlapping subsets are partially aligned.

Remark 1. As shown in [8], the capacity of PLC with individual privacy, which is a special case of IPLT for $L = 1$, is given by $\lceil \frac{K}{D+M} \rceil^{-1}$, where the user initially knows M uncoded messages or one linear combination of M messages as side information. The capacity of this setting was left open for $M = 0$. Theorem 1 provides a lower bound $(\lfloor \frac{K}{D} \rfloor + \min\{\frac{R}{S}, R\})^{-1}$ and an upper bound $(\lfloor \frac{K}{D} \rfloor + \min\{1, R\})^{-1}$ on the capacity of this setting. Interestingly, these bounds are matching when $R = 0$ or $R \mid D$, settling the capacity of PLC (without SI) with individual privacy for these cases. For $L = D$, IPLT reduces to PIR (without SI) with individual privacy, and an optimal scheme in this case is to download the entire dataset [13].

Remark 2. The result of Theorem 1 can be extended to IPLT with Side Information (SI). We consider two types of SI—previously studied in the PIR and PLC literature: *Uncoded SI (USI)* (see [15]), and *Coded SI (CSI)* (see [16]). In the case of USI (or CSI), the user initially knows a subset of M messages (or L MDS coded combinations of M messages). For both USI and CSI, the identities of these M messages are initially unknown by the server. When the identity of every message in the support sets of demand and side information must be protected individually, a slightly modified version of the GPC-PIA scheme (for IPLT without SI) achieves the rate $(\lfloor \frac{K}{D+M} \rfloor + \min\{\frac{R}{S}, \frac{R}{L}\})^{-1}$ for both IPLT with USI and CSI, where $R = K \pmod{D+M}$ and $S = \gcd(D+M+R, R)$. This result generalizes the results of [13] and [8] for PIR and PLC with individual privacy. The optimality of this rate, however, remains open in general.

V. Proof of Converse

Lemma 2. *The rate of any IPLT protocol for K messages, demand's support size D and dimension L, is upper bounded by $(\lfloor \frac{K}{D} \rfloor + \min\{1, \frac{R}{L}\})^{-1}$, where $R \triangleq K \pmod{D}$.*

Proof: Consider an arbitrary IPLT protocol that generates the query-answer pair $(\mathrm{Q}^{[\mathrm{W},\mathrm{V}]}, \mathrm{A}^{[\mathrm{W},\mathrm{V}]})$ for any given (W, V). To prove the rate upper bound in the lemma, we need to show that $H(\mathbf{A}) \geq (L\lfloor \frac{K}{D} \rfloor + \min\{L, R\})\theta$. Recall that $\mathbf{A}$ denotes $\mathbf{A}^{[\mathbf{W},\mathbf{V}]}$, and θ is the entropy of a message. Consider an arbitrary message index $k_1 \in \mathcal{K}$. By the result of Lemma 1, there exist $\mathrm{W}_1 \in \mathcal{W}$ with $k_1 \in \mathrm{W}_1$, and $\mathrm{V}_1 \in \mathcal{V}$ such that $H(\mathbf{Z}_1|\mathbf{A}, \mathbf{Q}) = 0$, where $\mathbf{Z}_1 \triangleq \mathbf{Z}^{[\mathrm{W}_1,\mathrm{V}_1]}$. By the same arguments as in the proof of [2, Lemma 2], we have

$$\begin{aligned} H(\mathbf{A}) &\geq H(\mathbf{A}|\mathbf{Q}) + H(\mathbf{Z}_1|\mathbf{A}, \mathbf{Q}) \\ &= H(\mathbf{Z}_1|\mathbf{Q}) + H(\mathbf{A}|\mathbf{Q}, \mathbf{Z}_1) \\ &= H(\mathbf{Z}_1) + H(\mathbf{A}|\mathbf{Q}, \mathbf{Z}_1) \end{aligned} \tag{1}$$

To further lower bound $H(\mathbf{A}|\mathbf{Q}, \mathbf{Z}_1)$, we proceed as follows. Take an arbitrary message index $k_2 \notin \mathrm{W}_1$. Again, by Lemma 1, there exist $\mathrm{W}_2 \in \mathcal{W}$ with $k_2 \in \mathrm{W}_2$, and $\mathrm{V}_2 \in \mathcal{V}$ such that $H(\mathbf{Z}_2|\mathbf{A}, \mathbf{Q}) = 0$, where $\mathbf{Z}_2 \triangleq \mathbf{Z}^{[\mathrm{W}_2,\mathrm{V}_2]}$. Using a similar technique as in (1), it follows that $H(\mathbf{A}|\mathbf{Q}, \mathbf{Z}_1) \geq H(\mathbf{Z}_2|\mathbf{Q}, \mathbf{Z}_1) + H(\mathbf{A}|\mathbf{Q}, \mathbf{Z}_1, \mathbf{Z}_2)$, and consequently,

$$H(\mathbf{A}|\mathbf{Q}, \mathbf{Z}_1) \geq H(\mathbf{Z}_2|\mathbf{Z}_1) + H(\mathbf{A}|\mathbf{Q}, \mathbf{Z}_2, \mathbf{Z}_1). \tag{2}$$

Combining (1) and (2), we get

$$H(\mathbf{A}) \geq H(\mathbf{Z}_1) + H(\mathbf{Z}_2|\mathbf{Z}_1) + H(\mathbf{A}|\mathbf{Q}, \mathbf{Z}_2, \mathbf{Z}_1). \tag{3}$$

We repeat this lower-bounding process multiple rounds until there is no message index left to take. Let n be the total number of rounds, and let $k_1, \ldots, k_n$ be the message indices chosen over the rounds. For every $i \in \{1, \ldots, n\}$, let $\mathrm{W}_i \in \mathcal{W}$ with $k_i \in \mathrm{W}_i$ and $k_i \notin \cup_{1 \leq j < i} \mathrm{W}_j$, and $\mathrm{V}_i \in \mathcal{V}$, be such that $H(\mathbf{Z}_i|\mathbf{A}, \mathbf{Q}) = 0$, where $\mathbf{Z}_i \triangleq \mathbf{Z}^{[\mathrm{W}_i,\mathrm{V}_i]}$. (For any $i \in \{1, \ldots, n\}$, W_i and V_i exist due to Lemma 1.) Note that $\cup_{1 \leq i \leq n} \mathrm{W}_i = \mathcal{K}$. Similarly as before, we can show that

$$\begin{aligned} H(\mathbf{A}) &\geq \sum_{1 \leq i \leq n} H(\mathbf{Z}_i|\mathbf{Z}_{i-1}, \ldots, \mathbf{Z}_1) + H(\mathbf{A}|\mathbf{Q}, \mathbf{Z}_n, \ldots, \mathbf{Z}_1) \\ &\geq \sum_{1 \leq i \leq n} H(\mathbf{Z}_i|\mathbf{Z}_{i-1}, \ldots, \mathbf{Z}_1). \end{aligned} \tag{4}$$

Next, we show that

$$H(\mathbf{Z}_i|\mathbf{Z}_{i-1}, \ldots, \mathbf{Z}_1) \geq \min\{N_i, L\}\theta, \tag{5}$$

where $N_i \triangleq |\mathrm{W}_i \setminus \cup_{1 \leq j < i} \mathrm{W}_j|$ is the number of message indices that belong to W_i, but not $\cup_{1 \leq j < i} \mathrm{W}_j$. (Note that $N_1 = |\mathrm{W}_1| = D$.) Let $\mathbf{Z}_{i,1}, \ldots, \mathbf{Z}_{i,L}$ be the (row-) vectors pertaining to $\mathbf{Z}_i$, where $\mathbf{Z}_{i,l} \triangleq \mathrm{v}_{i,l} \mathbf{X}_{\mathrm{W}_i}$, and $\mathrm{v}_{i,l}$ is the lth row of V_i. Note that these vectors are linear combinations of the messages $\mathbf{X}_1, \ldots, \mathbf{X}_K$. We need to show that there exist $M_i \triangleq \min\{N_i, L\}$ vectors pertaining to $\mathbf{Z}_i$ that are independent of the vectors pertaining to $\mathbf{Z}_1, \ldots, \mathbf{Z}_{i-1}$. Let $\mathrm{u}_{i,l}$ be a row-vector of length K such that the vector $\mathrm{u}_{i,l}$ restricted to its components indexed by W_i is the vector $\mathrm{v}_{i,l}$, and the rest of the components of the vector $\mathrm{u}_{i,l}$ are all zero, and let $\mathrm{U}_i \triangleq [\mathrm{u}_{i,1}^{\mathsf{T}}, \ldots, \mathrm{u}_{i,L}^{\mathsf{T}}]^{\mathsf{T}}$. Thus, we need to show that the matrix U_i contains M_i rows that are linearly independent of the rows of the matrices $\mathrm{U}_1, \ldots, \mathrm{U}_{i-1}$. Note that the rows of the matrix U_i are linearly independent, because U_i contains V_i as a submatrix, and V_i is invertible. Let S_i be an $L \times N_i$ submatrix of U_i formed by columns indexed by $\mathrm{W}_i \setminus \cup_{1 \leq j < i} \mathrm{W}_j$. Note that S_i is a submatrix of V_i, and every $L \times L$ submatrix of V_i is invertible. We consider two cases: (i) $N_i \leq L$, and (ii) $N_i > L$. In the

case (i), the N_i columns of S_i are linearly independent. Otherwise, any $L\times L$ submatrix of V_i that contains S_i cannot be invertible, and hence a contradiction. In the case (ii), any L columns of S_i are linearly independent. Otherwise, S_i (and V_i) contains an $L\times L$ submatrix that is not invertible, which is a contradiction. By these arguments, the rank of S_i is $M_i=\min\{L,N_i\}$, and S_i contains M_i linearly independent rows. Without loss of generality, assume that the first M_i rows of S_i are linearly independent. Moreover, the submatrix of $[\mathrm{U}_1^{\mathsf{T}},\ldots,\mathrm{U}_{i-1}^{\mathsf{T}}]^{\mathsf{T}}$ restricted to its columns indexed by $\mathrm{W}_i\setminus\cup_{1\le j<i}\mathrm{W}_j$ (and all its rows) is an all-zero matrix. Thus, the first M_i rows of U_i are linearly independent of the rows of $[\mathrm{U}_1^{\mathsf{T}},\ldots,\mathrm{U}_{i-1}^{\mathsf{T}}]^{\mathsf{T}}$. This completes the proof of (5).

Combining (4) and (5), we have

$$H(\mathbf{A})\ge\sum_{1\le i\le n}\min\{L,N_i\}\theta \tag{6}$$

Recall that $N_i=|\mathrm{W}_i\setminus\cup_{1\le j<i}\mathrm{W}_j|$. Note that $1\le N_i\le D$ since $\mathrm{W}_i\setminus\cup_{1\le j<i}\mathrm{W}_j$ is a subset of W_i, and the message index k_i belongs to $\mathrm{W}_i\setminus\cup_{1\le j<i}\mathrm{W}_j$. Moreover, $\sum_{i=1}^n N_i=K$ since W_1, $\mathrm{W}_2\setminus\mathrm{W}_1$, $\ldots$, $\mathrm{W}_n\setminus\cup_{1\le j<n}\mathrm{W}_j$ form a partition of $\mathcal{K}$, and $|\mathrm{W}_1|=N_1=D$, $|\mathrm{W}_2\setminus\mathrm{W}_1|=N_2$, $\ldots$, $|\mathrm{W}_n\setminus\cup_{1\le j<n}\mathrm{W}_j|=N_n$. To obtain a converse bound, we need to minimize $\sum_{1\le i\le n}\min\{L,N_i\}$, subject to the constraints (i) $N_1=D$, and $1\le N_i\le D$ for any $1<i\le n$, and (ii) $\sum_{1\le i\le n}N_i=K$. To this end, we reformulate this optimization problem as follows. For every $j\in\{1,\ldots,D\}$, let $T_j\triangleq\sum_{1\le i\le n}\mathbb{1}_{\{N_i=j\}}$ be the number of rounds i such that $N_i=j$. Using this notation, the objective function $\sum_{1\le i\le n}\min\{L,N_i\}$ can be rewritten as $\sum_{1\le j\le D}T_j\min\{L,j\}$, or equivalently, $\sum_{1\le j\le L}T_jj+\sum_{L<j\le D}T_jL$; the constraint (i) reduces to $T_j\in\mathbb{N}_0\triangleq\{0,1,\ldots\}$ for every $1\le j<D$, and $T_D\in\mathbb{N}\triangleq\{1,2,\ldots\}$; and the constraint (ii) reduces to $\sum_{1\le j\le D}T_jj=K$. Thus, we need to solve the following integer linear programming (ILP) problem:

$$\begin{aligned}\text{minimize}\quad&\sum_{1\le j\le L}T_jj+\sum_{L<j\le D}T_jL\\ \text{subject to}\quad&\sum_{1\le j\le D}T_jj=K\\ &T_1,\ldots,T_{D-1}\in\mathbb{N}_0,T_D\in\mathbb{N}\end{aligned}$$

Solving this ILP using the Gomory's cutting-plane algorithm [17], an optimal solution is $T_D=\lfloor\frac{K}{D}\rfloor$, $T_R=1$, and $T_j=0$ for all $j\notin\{R,D\}$, where $R\triangleq K\pmod D$, and the optimal value is $L\lfloor\frac{K}{D}\rfloor+\min\{L,R\}$. Equivalently, $\sum_{i=1}^n\min\{L,N_i\}\ge L\lfloor\frac{K}{D}\rfloor+\min\{L,R\}$. Combining this inequality and the inequality (6), we have $H(\mathbf{A})\ge(L\lfloor\frac{K}{D}\rfloor+\min\{L,R\})\theta$, as was to be shown. □

VI. Achievability Scheme

This section presents an IPLT protocol, called *Generalized Partition-and-Code with Partial Interference Alignment (GPC-PIA)*, that achieves the rate $(\lfloor\frac{K}{D}\rfloor+\min\{\frac{R}{S},\frac{R}{L}\})^{-1}$, where $R\triangleq K\pmod D$ and $S\triangleq\gcd(D+R,R)$. Examples of this protocol—not presented here due to space constraints—can be found in [1].

In the following, we denote by $\tilde{\mathrm{W}}$ a sequence of length D (instead of a set of size D) that the user initially constructs by randomly permuting the elements in the demand's support index set W, and denote by $\tilde{\mathrm{V}}$ an $L\times D$ matrix that the user initially constructs by applying the same permutation on the columns of the demand's coefficient matrix V.

We consider two different cases: (i) $L\le S$, and (ii) $L>S$. In each case, the protocol consists of three steps as follows.

Step 1: The user constructs a matrix G and a permutation π, and sends them as the query $\mathrm{Q}^{[\mathrm{W},\mathrm{V}]}$ to the server. In the following, we describe the construction of the matrix G and the permutation π for the cases (i) and (ii) separately.

Case (i): Let $n\triangleq\lfloor\frac{K}{D}\rfloor-1$, $m\triangleq\frac{R}{S}+1$, and $t\triangleq\frac{D}{S}-1$. The user constructs an $L(n+m)\times K$ matrix G,

$$\mathrm{G}=\begin{bmatrix}\mathrm{G}_1&0&\ldots&0&0\\0&\mathrm{G}_2&\ldots&0&0\\\vdots&\vdots&\ddots&\vdots&\vdots\\0&0&\ldots&\mathrm{G}_n&0\\0&0&\ldots&0&\mathrm{G}_{n+1}\end{bmatrix} \tag{7}$$

where $\mathrm{G}_1,\ldots\mathrm{G}_n$ are $L\times D$ matrices, and G_{n+1} is an $Lm\times(D+R)$ matrix. The matrices (blocks) $\mathrm{G}_1,\ldots,\mathrm{G}_n,\mathrm{G}_{n+1}$ are constructed as follows.

The user randomly selects one of the blocks $\mathrm{G}_1,\ldots,\mathrm{G}_{n+1}$, where each of the blocks $\mathrm{G}_1,\ldots,\mathrm{G}_n$ is selected with probability $\frac{D}{K}$, and the block G_{n+1} is selected with probability $\frac{D+R}{K}$. Let i^* be the index of the selected block. Depending on the choice of i^*, the description of the protocol is different. In the following, we consider the cases of $1\le i^*\le n$ and $i^*=n+1$ separately.

First, consider the case of $1\le i^*\le n$. In this case, the user takes G_{i^*} to be the matrix $\tilde{\mathrm{V}}$, i.e., $\mathrm{G}_{i^*}=\tilde{\mathrm{V}}$. For any $i\in\{1,\ldots,n\}\setminus\{i^*\}$, the user takes G_i to be a randomly generated MDS matrix of size $L\times D$. The construction of G_{n+1} is as follows. First, the user randomly generates an MDS matrix C of size $L\times(D+R)$, and partitions the columns of C into $t+m$ column-blocks each of size $L\times S$, i.e., $\mathrm{C}=[\mathrm{C}_1,\ldots,\mathrm{C}_{t+m}]$. Then, the user constructs $\mathrm{G}_{n+1}=[\mathrm{B}_1,\mathrm{B}_2]$, where B_1 and B_2 are given by

$$\begin{bmatrix}\alpha_1\omega_{1,1}\mathrm{C}_1&\ldots&\alpha_t\omega_{1,t}\mathrm{C}_t\\\vdots&\vdots&\vdots\\\alpha_1\omega_{m,1}\mathrm{C}_1&\ldots&\alpha_t\omega_{m,t}\mathrm{C}_t\end{bmatrix}\text{ and }\begin{bmatrix}\alpha_{t+1}\mathrm{C}_{t+1}&&\\&\ddots&\\&&\alpha_{t+m}\mathrm{C}_{t+m}\end{bmatrix}$$

respectively, and $\alpha_1,\ldots,\alpha_{t+m}$ are $t+m$ randomly chosen elements from $\mathbb{F}_p\setminus\{0\}$, and $\omega_{i,j}\triangleq(x_i-y_j)^{-1}$ for $1\le i\le m$ and $1\le j\le t$, where $x_1,\ldots,x_m$ and $y_1,\ldots,y_t$ are $t+m$ distinct elements chosen at random from $\mathbb{F}_p$. Note that $\omega_{i,j}$ is the entry (i,j) of an $m\times t$ Cauchy matrix.

Now, consider the case of $i^*=n+1$. For any $i\in\{1,\ldots,n\}$, the user takes G_i to be a randomly generated MDS matrix of size $L\times D$. The user then constructs G_{n+1} with a structure similar to that in the previous case, but for a different choice of matrices $\mathrm{C}_1,\ldots,\mathrm{C}_{t+m}$ and parameters $\alpha_1,\ldots,\alpha_{t+m}$, as specified below.

First, the user partitions the columns of $\tilde{\mathrm{V}}$ into $t+1$ column-blocks each of size $L\times S$, i.e., $\tilde{\mathrm{V}}=[\tilde{\mathrm{V}}_1,\ldots,\tilde{\mathrm{V}}_{t+1}]$.

The user then randomly chooses $t+1$ indices from $\{1,\ldots,t+m\}$, say, $i_1,\ldots,i_{t+1}$, and for any $1\le j\le t+1$, takes $\mathrm{C}_{i_j}=\tilde{\mathrm{V}}_j$. Next, the user randomly generates the rest of C_i's such that $\mathrm{C}=[\mathrm{C}_1,\ldots,\mathrm{C}_{t+m}]$ is an MDS matrix. The choice of α_i's is described below.

Hereafter, we refer to the submatrix of G_{n+1} formed by the ith L rows as the ith row-block of G_{n+1}. Note that G_{n+1} has m row-blocks. Let s be the number of column-block indices i_j for $j\in\{1,\ldots,t+m\}$ such that $i_j>t$. Note that $\mathrm{C}_{i_1},\ldots,\mathrm{C}_{i_{t-s+1}}$ belong to the matrix B_1, and $\mathrm{C}_{i_{t-s+2}},\ldots,\mathrm{C}_{i_{t+1}}$ belong to the matrix B_2. Let $\mathcal{I}\triangleq\{i_1,\ldots,i_{t+1}\}$ be the index set of those column-blocks of C that correspond to the column-blocks of $\tilde{\mathrm{V}}$. Let $\mathcal{I}_1\triangleq\{i_1,\ldots,i_{t-s+1}\}$, and let $\mathcal{I}_2\triangleq\mathcal{I}\backslash\mathcal{I}_1$. Note that for any $i\in\mathcal{I}_1$, C_i appears in all row-blocks of G_{n+1}, and for any $i\in\mathcal{I}_2$, C_i appears only in the $(i-t)$th row-block of G_{n+1}. The parameters α_i's are to be chosen such that, by performing row-block operations on G_{n+1}, the user can construct an $L\times(D+R)$ matrix with $t+m$ column-blocks each of size $L\times S$ that satisfies the following two conditions: (a) the blocks indexed by $\{1,\ldots,t+m\}\setminus\mathcal{I}$ are all zero, and (b) the blocks indexed by $\mathcal{I}=\{i_1,\ldots,i_{t+1}\}$ are $\mathrm{C}_{i_1},\ldots,\mathrm{C}_{i_{t+1}}$. For simplifying the notation, let $\{j_1,\ldots,j_{s-1}\}\triangleq\{1,\ldots,t\}\setminus\mathcal{I}$, and let $\{k_1,\ldots,k_s\}\triangleq\mathcal{I}_2=\{i_{t-s+2},\ldots,i_{t+1}\}$.

To perform row-block operations, for every $i\in\mathcal{I}_2=\{k_1,\ldots,k_s\}$, the user multiplies the $(i-t)$th row-block of G_{n+1} by a nonzero coefficient c_i. Let $\mathrm{c}\triangleq[c_{k_1},\ldots,c_{k_s}]^{\mathsf{T}}$. Upon choosing $\alpha_{j_1},\ldots,\alpha_{j_{s-1}}$ randomly from $\mathbb{F}_p\setminus\{0\}$, it follows that the condition (a) is satisfied so long as $\mathrm{M}_1\mathrm{c}$ is an all-zero vector, where

$$\mathrm{M}_1\triangleq\begin{bmatrix}\omega_{k_1-t,j_1} & \omega_{k_2-t,j_1} & \cdots & \omega_{k_s-t,j_1}\\ \vdots & \vdots & \vdots & \vdots\\ \omega_{k_1-t,j_{s-1}} & \omega_{k_2-t,j_{s-1}} & \cdots & \omega_{k_s-t,j_{s-1}}\end{bmatrix}.$$

Since M_1 is a Cauchy matrix by the choice of $\omega_{i,j}$'s, the submatrix of M_1 formed by columns indexed by $\{2,\ldots,s\}$ is invertible [18]. Thus, for any arbitrary $c_{k_1}\neq 0$, there is a unique solution for the vector c. Note also that all the components of c are nonzero because every square submatrix of M_1 is invertible (by the properties of Cauchy matrices).

Given the vector c, the condition (b) is satisfied so long as $\alpha_{k_1}=1/c_{k_1},\ldots,\alpha_{k_s}=1/c_{k_s}$, and $\alpha_{i_1},\ldots,\alpha_{i_{t-s+1}}$ are chosen such that $\mathrm{M}_2\mathrm{c}$ is an all-one vector, where

$$\mathrm{M}_2\triangleq\begin{bmatrix}\alpha_{i_1}\omega_{k_1-t,i_1} & \cdots & \alpha_{i_1}\omega_{k_s-t,i_1}\\ \vdots & \vdots & \vdots\\ \alpha_{i_{t-s+1}}\omega_{k_1-t,i_{t-s+1}} & \cdots & \alpha_{i_{t-s+1}}\omega_{k_s-t,i_{t-s+1}}\end{bmatrix}.$$

Solving for $\alpha_{i_1},\ldots,\alpha_{i_{t-s+1}}$, it follows that $\alpha_{i_j}\triangleq(\sum_{1\le l\le s}c_{k_l}\omega_{k_l-t,i_j})^{-1}$ for $1\le j\le t-s+1$. Note that $\alpha_{i_1},\ldots,\alpha_{i_{t-s+1}}$ are nonzero. Note also that $\sum_{1\le l\le s}c_{k_l}\omega_{k_l-t,i_j}$ is nonzero because the jth row of M_2 is linearly independent of the rows of M_1. For any $i\in\{1,\ldots,t+m\}\setminus\{i_1,\ldots,i_{t+1},j_1,\ldots,j_{s-1}\}$, the user chooses α_i randomly from $\mathbb{F}_p\setminus\{0\}$. This completes the construction of the matrix G.

Next, the user constructs a permutation π as follows. Let $\tilde{\mathrm{W}}=\{l_1,\ldots,l_D\}$, and let $\mathcal{K}\setminus\mathrm{W}=\{l_{D+1},\ldots,l_K\}$. First, consider the case of $1\le i^*\le n$. In this case, the user constructs π such that: for every $1\le j\le D$, $\pi(l_j)=(i^*-1)D+j$; and for every $D<j\le K$, $\pi(l_j)$ is a randomly chosen element from $\mathcal{K}\setminus\{\pi(l_k)\}_{1\le k<j}$. Next, consider the case of $i^*=n+1$. Recall that $i_1,\ldots,i_{t+1}$ are the indices of those column-blocks of C that correspond to the column-blocks of $\tilde{\mathrm{V}}$. Thus, the user constructs π such that: for every $1\le k\le t+1$ and $(k-1)S+1\le j\le kS$, $\pi(l_j)=nD+(i_k-1)S+f_j$, where $f_j=j\pmod S$ if $S\nmid j$, and $f_j=S$ if $S\mid j$; and for every $D<j\le K$, $\pi(l_j)$ is a randomly chosen element from $\mathcal{K}\setminus\{\pi(l_k)\}_{1\le k<j}$.

Case (ii): Let $n\triangleq\lfloor\frac{K}{D}\rfloor-1$, and $m\triangleq\frac{R}{L}+1$. The user constructs an $L(n+m)\times K$ matrix G with a structure similar to (7), where $\mathrm{G}_1,\ldots,\mathrm{G}_n$ are constructed similarly as in the previous case, but the construction of G_{n+1} is different. In the following, we will only explain how to construct G_{n+1}.

For the case of $1\le i^*\le n$, the user randomly generates an $[D+R,L+R]$ MDS code, and takes G_{n+1} to be the generator matrix of this code. For the case of $i^*=n+1$, the user constructs a $[D+R,L+R]$ MDS code using the same technique as in the step 1 of the Specialized MDS Code protocol of [2], except where K is replaced by $D+R$, and W is replaced by a randomly chosen D-subset of $\{1,\ldots,D+R\}$, say, $\{h_1,\ldots,h_D\}$. The user then uses the generator matrix of the constructed MDS code as G_{n+1}.

Next, the user constructs a permutation π. For the case of $1\le i^*\le n$, π is generated exactly the same as in the case (i), whereas the construction of π for the case of $i^*=n+1$ is different from that in the case (i). Similarly as before, let $\tilde{\mathrm{W}}=\{l_1,\ldots,l_D\}$, and let $\mathcal{K}\setminus\mathrm{W}=\{l_{D+1},\ldots,l_K\}$. For the case of $i^*=n+1$, the user constructs π such that: for every $1\le j\le D$, $\pi(l_j)=nD+h_j$; and for every $D<j\le K$, $\pi(l_j)$ is a randomly chosen element from $\mathcal{K}\setminus\{\pi(l_k)\}_{1\le k<j}$.

Step 2: Given the query $\mathrm{Q}^{[\mathrm{W},\mathrm{V}]}$, i.e., the matrix G and the permutation π, the server first constructs the matrix $\tilde{\mathrm{X}}\triangleq\pi(\mathrm{X})$ by permuting the rows of the matrix X according to the permutation π, i.e., for every $l\in\mathcal{K}$, $\pi(l)$th row of $\tilde{\mathrm{X}}$ is the lth row of X. Then, the server computes the matrix $\mathrm{Y}\triangleq\mathrm{G}\tilde{\mathrm{X}}$, and sends Y back to the user as the answer $\mathrm{A}^{[\mathrm{W},\mathrm{V}]}$.

Step 3: Upon receiving the answer $\mathrm{A}^{[\mathrm{W},\mathrm{V}]}$, i.e., the matrix Y, the user recovers the demand $\mathrm{Z}^{[\mathrm{W},\mathrm{V}]}$ as follows. For every $1\le i\le n$, let Y_i be the matrix Y restricted to its rows indexed by $\{(i-1)L+1,\ldots,iL\}$, and let Y_{n+1} be the matrix Y restricted to its rows indexed by $\{nL+1,\ldots,nL+mL\}$. For the case of $1\le i^*\le n$, $\mathrm{Z}^{[\mathrm{W},\mathrm{V}]}$ can be recovered from the matrix Y_{i^*} for both cases (i) and (ii). For the case of $i^*=n+1$, $\mathrm{Z}^{[\mathrm{W},\mathrm{V}]}$ can be recovered by performing proper row-block or row operations on the augmented matrix $[\mathrm{G}_{n+1},\mathrm{Y}_{n+1}]$ for the case (i) or (ii), respectively.

Lemma 3. *The GPC-PIA protocol is an IPLT protocol, and achieves the rate* $(\lfloor\frac{K}{D}\rfloor+\min\{\frac{R}{S},\frac{R}{L}\})^{-1}$*, where* $R\triangleq K\pmod D$ *and* $S\triangleq\gcd(D+R,R)$.

Proof: The proof can be found in [1]. □

REFERENCES

[1] N. Esmati, A. Heidarzadeh, and A. Sprintson, "Private linear transformation: The individual privacy case," Feb 2021. [Online]. Available: arXiv:2102.01662

[2] ——, "Private linear transformation: The joint privacy case," Feb 2021. [Online]. Available: arXiv:2102.01665

[3] K. Banawan and S. Ulukus, "Multi-message private information retrieval: Capacity results and near-optimal schemes," *IEEE Transactions on Information Theory*, vol. 64, no. 10, pp. 6842–6862, Oct 2018.

[4] A. Heidarzadeh, S. Kadhe, B. Garcia, S. E. Rouayheb, and A. Sprintson, "On the capacity of single-server multi-message private information retrieval with side information," in *2018 56th Annual Allerton Conf. on Commun., Control, and Computing*, Oct 2018.

[5] S. Li and M. Gastpar, "Single-server multi-message private information retrieval with side information," in *2018 56th Annual Allerton Conf. on Commun., Control, and Computing*, Oct 2018.

[6] M. H. Mousavi, M. Ali Maddah-Ali, and M. Mirmohseni, "Private inner product retrieval for distributed machine learning," in *2019 IEEE International Symposium on Information Theory (ISIT)*, 2019, pp. 355–359.

[7] A. Heidarzadeh and A. Sprintson, "Private computation with side information: The single-server case," in *2019 IEEE International Symposium on Information Theory (ISIT)*, July 2019, pp. 1657–1661.

[8] ——, "Private computation with individual and joint privacy," in *2020 IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 1112–1117.

[9] H. Sun and S. A. Jafar, "The capacity of private computation," *IEEE Transactions on Information Theory*, vol. 65, no. 6, pp. 3880–3897, 2019.

[10] M. Mirmohseni and M. A. Maddah-Ali, "Private function retrieval," in *2018 Iran Workshop on Communication and Information Theory (IWCIT)*, April 2018, pp. 1–6.

[11] S. A. Obead and J. Kliewer, "Achievable rate of private function retrieval from MDS coded databases," *2018 IEEE International Symposium on Information Theory (ISIT)*, pp. 2117–2121, 2018.

[12] S. A. Obead, H.-Y. Lin, E. Rosnes, and J. Kliewer, "Capacity of private linear computation for coded databases," *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 813–820, 2018.

[13] A. Heidarzadeh, S. Kadhe, S. E. Rouayheb, and A. Sprintson, "Single-server multi-message individually-private information retrieval with side information," in *2019 IEEE International Symposium on Information Theory (ISIT)*, July 2019, pp. 1042–1046.

[14] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: Applications to image and text data," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '01. New York, NY, USA: Association for Computing Machinery, 2001, p. 245–250. [Online]. Available: https://doi.org/10.1145/502512.502546

[15] S. Kadhe, B. Garcia, A. Heidarzadeh, S. El Rouayheb, and A. Sprintson, "Private information retrieval with side information," *IEEE Transactions on Information Theory*, vol. 66, no. 4, pp. 2032–2043, 2020.

[16] A. Heidarzadeh, F. Kazemi, and A. Sprintson, "The role of coded side information in single-server private information retrieval," *IEEE Transactions on Information Theory*, vol. 67, no. 1, pp. 25–44, 2021.

[17] H. Marchand, A. Martin, R. Weismantel, and L. Wolsey, "Cutting planes in integer and mixed integer programming," *Discrete Applied Mathematics*, vol. 123, no. 1, pp. 397 – 446, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0166218X01003481

[18] R. Roth, *Introduction to Coding Theory*. New York, NY, USA: Cambridge University Press, 2006.