Multi-Agent Distributed Optimization With Feasible Set Privacy*

Shreya Meel Sennur Ulukus

Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742
smeel@umd.edu *ulukus@umd.edu*

Abstract

We consider the problem of decentralized constrained optimization with multiple agents $E_1, \ldots, E_N$ who jointly wish to learn the optimal solution set while keeping their feasible sets $\mathcal{P}_1, \ldots, \mathcal{P}_N$ private from each other. We assume that the objective function f is known to all agents and each feasible set is a collection of points from a universal alphabet $\mathcal{P}_{alph}$. A designated agent (leader) starts the communication with the remaining (non-leader) agents, and is the first to retrieve the solution set. The leader searches for the solution by sending queries to and receiving answers from the non-leaders, such that the information on the individual feasible sets revealed to the leader should be no more than nominal, i.e., what is revealed from learning the solution set alone. We develop achievable schemes for obtaining the solution set at nominal information leakage, and characterize their communication costs under two communication setups between agents. In this work, we focus on two kinds of network setups: i) ring, where each agent communicates with two adjacent agents, and ii) star, where only the leader communicates with the remaining agents. We show that, if the leader first learns the joint feasible set through an existing private set intersection (PSI) protocol and then deduces the solution set, the information leaked to the leader is greater than nominal. Moreover, we draw connection of our schemes to threshold PSI (ThPSI), which is a PSI-variant where the intersection is revealed only when its cardinality is larger than a threshold value. Finally, for various realizations of f mapped uniformly at random to a fixed range of values, our schemes are more communication-efficient with a high probability compared to retrieving the entire feasible set through PSI.

*This work was presented in part at CISS 2024.

1 Introduction

In distributed optimization, agents collaborate to find the optimal solution of a global objective function. Each agent has its own feasible set and the solution of the constrained optimization problem should be present in the intersection of all feasible sets as shown in Fig. 1. The feasible set contains sensitive information of an agent, and should be kept private. For instance, consider the problem of allocating charging schedules to electric vehicles (EVs) [1], to minimize load fluctuations on a power grid subject to maximum charge and energy constraints for each EV. Then, the maximum charge for an EV being zero suggests that the owner of the EV (agent) is away, leaking private information about the agent. Most importantly, the iterative optimization algorithms [2] involve exchanging solution estimates among agents over multiple rounds which, on collusion of all but one agent reveals information on the target agent's feasible set, breaching its privacy.

As a solution, differential privacy (DP) [3] based optimization algorithms alleviate this problem to some extent by adding noise to the estimates before exchanging them. For instance, [1] incorporates DP in their algorithm to protect the privacy of agents' information in their respective constraints. This procedure retains some uncertainty on the agents' feasible sets because of the noise-added estimates observed by the participating agents. Besides feasible sets, iterative distributed algorithms risk the privacy of agents' local functions as well. The global objective is given by the sum of these local functions and guaranteeing DP to agents in these scenarios was studied in [4–7]. DP-based guarantees to protect user information in the objective function and constraints of a linear program was studied in [8]. However, the main disadvantage of DP-based approaches is that they compromise the accuracy of solutions owing to the noise added. Moreover, they allow some certainty in an adversary's guess of the agent's private information, unlike full uncertainty as guaranteed by information-theoretic privacy. To achieve accurate solution, homomorphic encryption schemes can be incorporated in the distributed algorithm as in [9]. However, the privacy in such schemes is provided by hardness in computing certain functions, which can be broken through infinite computing power.

To the best of our knowledge, none of the existing works address the problem of guaranteeing privacy to user's constraints in an information-theoretic sense. In this regard, we formulate distributed optimization with feasible set privacy (DOFSP), where the goal of the agents is to solve an optimization problem while keeping their feasible sets information-theoretically private from each other. For instance, in Fig. 1, the agents describe their feasible sets by a set of discrete points satisfying the linear inequality constraints. The feasible set of each agent is a finite list of values from a universal alphabet and the objective function f is known to all agents. The agents do not collude among themselves, and are assumed to be *honest but curious*, in the sense that they follow the communication protocol truthfully, but are curious to learn information on each other's private feasible sets. To search for the optimum set $\mathcal{P}^*$, one of the agents (*leader*) privately learns whether the best solutions in its

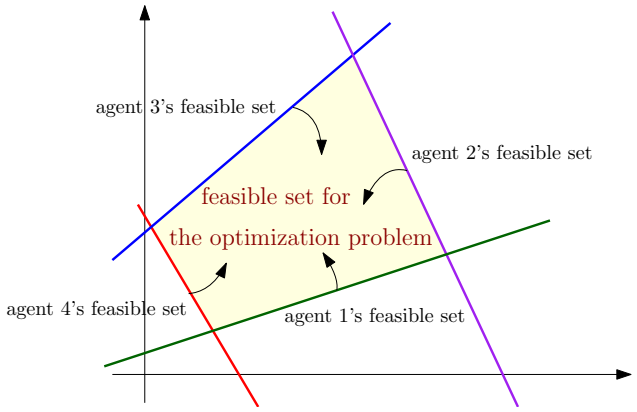
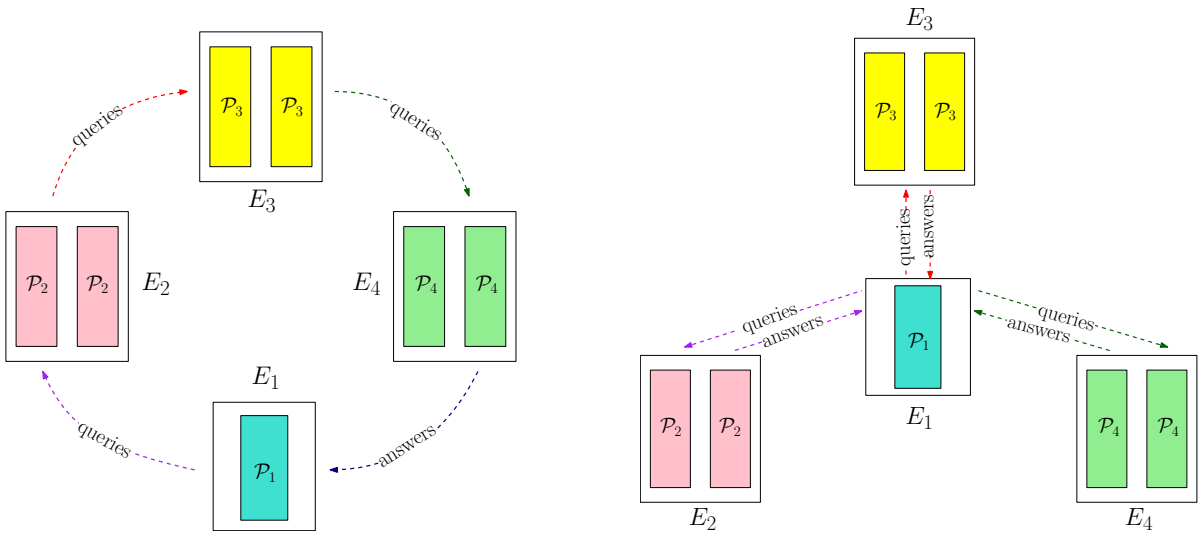


Figure 1: Feasible set for a distributed optimization problem with four agents each with a linear inequality constraint.

own feasible set are also present in the feasible sets of the remaining (*non-leader*) agents, while learning *no further information* on their individual feasible sets. In fact, the agents learn no more information on the intersection of the feasible sets, than what is revealed from learning the solution set alone. We call this information leakage as *nominal information leakage*.

The agents achieve this by first mapping the feasible sets into incidence vectors [10, 11], storing them on non-colluding databases and exchanging carefully designed messages among themselves till the optimum solution set $\mathcal{P}^*$ is learned by a designated agent, i.e., leader. Our protocols require the incidence vectors of feasible sets to be replicatedly stored in at least two non-colluding databases of the non-leader agents. This is along the lines of the information-theoretic formulation of private information retrieval (PIR) [12–16] where messages are replicated across multiple databases. In PIR, a user wants to retrieve one of these messages, without revealing the identity of the message to an individual server. Specifically, our protocols are derived from symmetric PIR (SPIR) schemes [17]. In SPIR, the additional requirement beyond PIR is that the messages in the databases, other than the desired one, should be hidden from the retrieving user.

For the simple two-agent case, our scheme proceeds as follows. The leader privately checks if the *best* solutions in the leader's set also appear in the other agent's set, by invoking the cardinality PSI (CarPSI) protocol which returns the number of such elements. If none of the candidate elements are present in the intersection, i.e., CarPSI returns a zero, the agent re-invokes CarPSI, this time with the *next best* set of elements. Otherwise, the solution appears on both feasible sets, hence at the intersection, giving the optimum function value and the leader learns the number of such elements. Next, the leader finds $\mathcal{P}^*$ by executing FindPSI protocol, which privately finds the set intersection, with the knowledge of its cardinality. Alternatively, one can view the DOFSP scheme as a sequential application of a two-user *threshold private set intersection* (ThPSI) protocol wherein the set intersection is evaluated only if its cardinality is greater than or equal to a threshold, while revealing no



(a) Ring setup. (b) Star setup.

Figure 2: Two types of communication setups for $N = 4$ entities. Leader entity E_1 has a single database, each non-leader entity, E_2, E_3, E_4, has $N_i = 2$ databases, $i = 2, 3, 4$.

further information about the sets as in PSI [18]. As an example, an important instance of ThPSI arises in a ride sharing app where each passenger wants to share information on their trajectory only when there is sufficient overlap between the routes. In the context of our scheme, the ThPSI operates on subsets dictated by the function values and the threshold is 1 in our case. Hence, we obtain a ThPSI protocol guaranteeing perfect information-theoretic privacy unlike the existing protocols of [19, 20] which guarantee computational privacy only.

Next, we extend our scheme to the multi-agent scenario. In terms of design of the communication protocol, we consider two setups as shown in Fig. 2:

1. *Ring*, where each agent communicates with its two neighboring agents only. The leader sends queries to the next agent, who after processing them, sends queries to the next agent, and so on. The last agent sends answers back to the leader.

2. *Star*, where the leader communicates with the remaining $N - 1$ agents that do not communicate among themselves. The leader sends queries to the databases of the remaining agents, who respond to it with their answers.

We provide a DOFSP scheme for the ring setup by extending CarPSI and FindPSI to primitives CarPSI-ring and FindPSI-ring, respectively. The CarPSI-ring primitive starts with the leader sending shares [21] of its incidence vector to two non-colluding databases of the neighboring agent. Each database then uses the received message and the stored incidence vector and computes a message to be sent over the database of the next agent and so on. Finally, the leader collects the messages from the Nth database pair and computes the cardinality of the set intersection. If this is greater than or equal to the threshold value, the leader retrieves the set intersection by retrieving two additional messages from the Nth

agent by executing FindPSI-ring. For the star setup, our DOFSP scheme is a sequential application of the multi-party PSI scheme of [11].

The main contributions of this paper can be summarized as follows:

- We propose achievable DOFSP schemes for the two-user and multi-user setups, where we deal with two communication setups, ring and star in the latter case, and characterize their communication efficiency in terms of the upload and download costs, which are the number of symbols exchanged between the agents.
- We show that, unlike our DOFSP schemes, the naive approach of learning the feasible set first, and then evaluating $\mathcal{P}^*$ leaks more information than the nominal and incurs a higher or equal communication cost.
- We numerically evaluate the probability of the communication cost of our DOFSP scheme being equal to that of the naive scheme, and show that this probability is small. Specifically, under uniform mapping of f, we evaluate this probability for both the setups with $N \geq 2$ with a fixed feasible set assignment, and show through our numerical results that this probability is low.

The rest of the paper is organized as follows. Section 2 describes the DOFSP problem setting and Section 3 states the results of the paper. In Section 4, we present our DOFSP scheme with $N = 2$ agents. In Sections 5 and 6, we present the DOFSP schemes under the ring and star setups, respectively, for $N > 2$. Finally, we conclude the paper in Section 7.

Notations: To denote the set of integers $\{1, 2, \ldots, n\}$ for a positive integer n, we use the notation $[n]$. For positive integers m, n with $m < n$, the notation $[m : n]$ represents the set $\{m, m+1, \ldots, n-1, n\}$. We use the symbol $\mathbb{N}$ to denote the set of positive integers.

2 Problem Formulation

2.1 General Problem Setting

We consider a setting with N entities (agents) who jointly seek the optimal solution (or optimal solutions) of an objective function, subject to satisfying the constraints of all participating entities. Each entity E_i, $i \in [N]$, has a feasible set $\mathcal{P}_i$ with elements from a finite indexed alphabet set $\mathcal{P}_{alph}$ of size $|\mathcal{P}_{alph}| = K$. Therefore, each $\mathcal{P}_i$ of size P_i is a subset of $\mathcal{P}_{alph}$ and is private to E_i. For the existence of a non-empty solution set, we assume the feasible sets to have a non-empty intersection, i.e., $\mathcal{P} := \mathcal{P}_1 \cap \mathcal{P}_2 \cap \ldots \cap \mathcal{P}_N \neq \emptyset$.

Each E_i casts its feasible set $\mathcal{P}_i$ into an *incidence vector*, which is a K-length binary column vector X_i indicating the presence (or absence) of an element by 1 (or 0), respectively.

We write the kth entry, for $k \in [K]$, of X_i as follows,

$$X_i(k) = \begin{cases} 1, & \text{if } \mathcal{P}_{alph}(k) \in \mathcal{P}_i, \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where $\mathcal{P}_{alph}(k)$ is the element of $\mathcal{P}_{alph}$ at index k. Each entity E_i has access to $N_i \geq 2$ non-colluding databases, where each database of E_i stores the incidence vector X_i. The numbers P_i and N_i are known to all the entities for each $i \in [N]$.

The objective function f is modeled as a random mapping $f : \mathcal{P}_{alph} \to \mathcal{T}$ independent of the sets $\mathcal{P}_i$, where the set $\mathcal{T}$ takes τ distinct values. For our subsequent analysis, we assume that f maps each element in $\mathcal{P}_{alph}$ uniformly to an element in τ. In other words, for any $x \in \mathcal{P}_{alph}$ and $y \in \mathcal{T}$, $\mathbb{P}(f(x) = y) = \frac{1}{\tau}$. The mapping is done independently for each x.

The goal of $E_1, \ldots, E_N$ is to find the solution set $\mathcal{P}^*$ where f attains the optimum (minimum or maximum) value within the overall feasible set $\mathcal{P}$, while keeping their individual feasible sets private from each other. That is, for a minimization problem, the entities follow a communication protocol to find the solution to:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & x \in \mathcal{P}. \end{aligned} \tag{2}$$

However, the protocol should not reveal to E_i any information about the feasible sets other than its own, i.e., $\mathcal{P}_i$ beyond what is implied by $\mathcal{P}^*$. Thus, the associated information leakage on the remaining feasible sets to every agent should be *nominal* (minimal) which we define as follows.

Definition 1 (Nominal information leakage) *Nominal information leakage is the amount of information on the other feasible sets revealed to a specific entity by learning the solution set $\mathcal{P}^*$ only, given its own feasible set and the knowledge of f. For E_i, $i \in [N]$, it is denoted by $\mathcal{I}_{nom,i}$ and defined as*

$$\mathcal{I}_{nom,i} = I(\mathcal{P}^*; \mathcal{P}_1, \ldots, \mathcal{P}_{i-1}, \mathcal{P}_{i+1}, \ldots, \mathcal{P}_N | \mathcal{P}_i, f). \tag{3}$$

We assume that the communication protocol is started by a designated entity, *leader* (say, E_1). It is also the first entity that learns $\mathcal{P}^*$ and communicates it to the other entities. Given a realization of f, we define two vectors: Global function profile $\boldsymbol{\mu}$ and local function profile $\boldsymbol{\alpha}$ which indicate the number of elements in $\mathcal{P}_{alph}$ and $\mathcal{P}_1$, respectively, that map to the same function value via f.

1. **Global function profile:** Let $f_1, f_2, \ldots, f_T$ where $T \in [\tau]$ be ordered such that f_1 is the best and f_T is the worst function value in terms of the optimization objective.

Then, $\boldsymbol{\mu} = [\mu_1, \mu_2, \ldots, \mu_T]$, where $\mu_1 + \mu_2 + \ldots + \mu_T = K$, $\mu_t \geq 1$, $t \in [T]$. All entities have the knowledge of $\boldsymbol{\mu}$.

2. **Local function profile:** Let the equi-cost elements in $\mathcal{P}_1$ be ordered from best to worst as $f_{i_1}, f_{i_2}, \ldots, f_{i_L}$ where $i_1 < i_2 < \ldots < i_L$. Further, let $\mathcal{J}_r$ be the set of indices in X_1 corresponding to the rth best solution in $\mathcal{P}_1$, with f_{i_r} as the respective function value. Then, $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_L]$ with α_r denoting the multiplicity of the rth best solution, i.e., $|\mathcal{J}_r| = \alpha_r$ for all $r \in [L]$. Each $\alpha_r \geq 1$ and $\sum_{r=1}^{L} \alpha_r = P_1$. Only E_1 has the knowledge of $\mathcal{J}_{1:L}$ and $\boldsymbol{\alpha}$.

Let $\mathcal{M}'_{in}$ and $\mathcal{M}'_{out}$ denote the incoming and outgoing messages of E_1 in the communication protocol. Then, to ensure the correct recovery of $\mathcal{P}^*$ at E_1, we must have,

$$[\text{reliability at } E_1] \qquad H(\mathcal{P}^* | \mathcal{M}'_{out}, \mathcal{M}'_{in}, \mathcal{P}_1, f) = 0. \tag{4}$$

The leader E_1 should not gain, from the messages it sends and receives, any information on $\mathcal{P}_i, i \neq 1$, beyond the nominal leakage,

$$[\text{privacy of } E_{2:N} \text{ against } E_1] \qquad I(\mathcal{M}'_{in}, \mathcal{M}'_{out}; \mathcal{P}_{2:N} | \mathcal{P}_1, f) = \mathcal{I}_{nom,1}. \tag{5}$$

To ensure the privacy of the outgoing messages, the databases of each non-leader E_i, $i \in \{2, \ldots, N\}$, share a set of common randomness symbols, denoted by $\mathcal{S}_i$. Hence, the content stored at each database of E_i is X_i and $\mathcal{S}_i$. Let $\mathcal{M}_{in,i,j}$ and $\mathcal{M}_{out,i,j}$ denote the incoming and outgoing messages of database j of E_i. Then, the database learns no information about the remaining feasible sets, $\mathcal{P}_{[N]\setminus i}$,

$$[\text{privacy of } E_{[N]\setminus i} \text{ against database } j \text{ of } E_i] \qquad I(\mathcal{M}_{in,i,j}, \mathcal{M}_{out,i,j}; \mathcal{P}_{[N]\setminus i} | \mathcal{P}_i, \mathcal{S}_i) = 0. \tag{6}$$

In our communication protocol, the entities other than E_1 do not directly participate, and only their databases are involved in the respective communications. Therefore, it suffices if the individual databases do not learn any information about the other feasible sets. This ensures that the information leaked to any non-leader E_i is simply $\mathcal{I}_{nom,i}$.

An achievable scheme for distributed optimization with feasible set privacy (DOFSP), under a ring or star setup, should satisfy the reliability and privacy requirements in (4), (5) and (6). Next, we specify the messages sent and received in the protocol for each communication setup.

2.2 Ring Setup

Each non-leader entity E_i has $N_i \geq 2$ non-colluding databases. Let them be labeled as $1, 2, \ldots, N_i, \forall i$. We assume that there exists a noiseless, secure communication link between database j of E_i with database j of E_{i+1}, for all $i \in [2 : N-1]$. There also exist noiseless,

secure links between E_1 and the databases of E_2 and E_N. Without loss of generality, assume that databases in the set $\mathcal{N}_{eff} = \{1, \ldots, N_{eff}\}$ where $N_{eff} = \min_{i\in[2:N]} N_i$ form a secure ring across the entities, with $N_{eff} \geq 2$. Alternatively, only the databases in $\mathcal{N}_{eff}$ of every non-leader entity participate in the communication protocol.

The communication flow is as follows. E_1 sends queries $Q_{1,j}$ to database $j \in \mathcal{N}_{eff}$ of E_2. For the intermediate entities, $E_i, i \in [2 : N-1]$, each database j of E_i sends a query $Q_{i,j}$ to database j of E_{i+1}. Finally, database j of E_N sends its answer A_j to E_1. Thus, the communication flow can be described by a ring. The corresponding messages exchanged are

$$\mathcal{M}'_{out} = Q_{1,j}, \quad \mathcal{M}'_{in} = A_j, \quad j \in \mathcal{N}_{eff}. \tag{7}$$

Further, for the databases of $E_i, i \in [2 : N-1]$, we have

$$\mathcal{M}'_{out,i,j} = Q_{i,j}, \quad \mathcal{M}'_{in,i,j} = Q_{i-1,j}, \quad j \in \mathcal{N}_{eff}, \tag{8}$$

and for E_N, we have

$$\mathcal{M}'_{out,N,j} = A_j, \quad \mathcal{M}'_{in,N,j} = Q_{N-1,j}, \quad j \in \mathcal{N}_{eff}. \tag{9}$$

Each $Q_{i,j}$ is a function of $Q_{i-1,j}$, X_i and $\mathcal{S}_i$ and each A_j is a function of $Q_{N-1,j}, X_N$ and $\mathcal{S}_N$. The communication cost C_{ring} is the total number of symbols transmitted between entities till the protocol is completed.

2.3 Star Setup

In the star setup, we adopt the SPIR framework for PSI as in [10, 11]. Entity E_i has N_i databases where X_i is stored. The index $l^* \in [N]$ of the leader is assigned as

$$l^* = \arg\min_{l\in[N]} \sum_{i=1, i\neq l}^{N} \left\lceil \frac{P_l N_i}{N_i - 1} \right\rceil, \tag{10}$$

in order to minimize the communication cost of PSI. We relabel the entities to let $l^* = 1$. E_1 has a secure, noiseless communication link with each individual database of the non-leader entities. Apart from sharing the common randomness $\mathcal{S}_i$, the databases of $E_i, i \in [2 : N]$, do not collude among themselves. In the communication protocol, for $N \geq 2$, E_1 sends queries to and collects answers from the databases of the remaining entities. In particular, we have

$$\mathcal{M}'_{out} = \bigcup_{i\in[N]} \bigcup_{j\in[N_i]} Q_{i,j}, \quad \mathcal{M}'_{in} = \bigcup_{i\in[N]} \bigcup_{j\in[N_i]} A_{i,j}, \quad i \in \{2, \ldots, N\}, \tag{11}$$

where $Q_{i,j}$ is the query sent to and $A_{i,j}$ is the answer received from the jth database of E_i. Similarly,

$$\mathcal{M}_{in,i,j} = A_{i,j}, \quad \mathcal{M}_{out,i,j} = Q_{i,j}, \tag{12}$$

where each answer $A_{i,j}$ is a function of $Q_{i,j}$, X_i and $\mathcal{S}_i$. (For $N = 2$, we simplify $Q_{i,j}$ and $A_{i,j}$ to Q_j and A_j, respectively.) The download cost D_{star} refers to the total number of symbols of the operating field received by E_1 from all the non-leader databases. The upload cost U_{star} is the number of symbols sent by E_1 to these databases from start to end of the protocol. Their sum $C_{star} = D_{star} + U_{star}$ reflects the total communication cost. If $N = 2$, we denote these costs by C, D and U, respectively.

3 Main Results

Theorem 1 *For a $N = 2$ agent DOFSP problem, given the feasible sets $\mathcal{P}_1$ and $\mathcal{P}_2$, the worst-case download cost D depends on the position R of the best function value in X_1 and is given by,*

$$D = \begin{cases} \left\lceil \frac{(R+\alpha_R-1)N_2}{N_2-1} \right\rceil, & \alpha_R > X_{\mathcal{J}_R}^T X_2, \\ \left\lceil \frac{RN_2}{N_2-1} \right\rceil, & \alpha_R = X_{\mathcal{J}_R}^T X_2, \end{cases} \tag{13}$$

where $X_{\mathcal{J}_r} = \sum_{j \in \mathcal{J}_r} e_j$ and e_j is the K-length unit column vector with 1 at the jth position. The upload cost U for the above scheme is KD symbols, resulting in $C = U + D = (K+1)D$.

The proof of Theorem 1 follows from our proposed scheme in Section 4.

Theorem 2 *For an $N > 2$ agent DOFSP problem under the ring setup, given the feasible sets $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_N$, the worst-case communication cost C_{ring} is given by,*

$$C_{ring} = \begin{cases} 2N\left(\sum_{r=1}^{R} \mu_r\right) + 2\left(R - \sum_{r=1}^{R-1} \mu_r\right), & R < T, \\ 2NK + 2\left(R - 1 - \sum_{r=1}^{R-1} \mu_r\right), & R = T, \end{cases} \tag{14}$$

where $R \in [T]$ corresponds to the index of the optimum function value in $\boldsymbol{\mu}$.

Theorem 3 *For an $N > 2$ agent DOFSP problem under the star setup, given the feasible sets $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_N$, the worst case download cost D_{star} is given by,*

$$D_{star} = \sum_{i=2}^{N} \left\lceil \frac{(\sum_{r=1}^{R} \alpha_r) N_i}{N_i - 1} \right\rceil, \tag{15}$$

where $R \in [L]$ corresponds to the index of the optimum function value in $\boldsymbol{\alpha}$. The upload cost U_{star} for the above scheme is KD_{star}, giving $C_{star} = (K+1)D_{star}$.

The proofs of Theorems 2 and 3 follow from the schemes in Sections 5 and 6, respectively.

4 The Case of $N = 2$ Entities

When $N = 2$, we refer to the leader E_1 as *client* and E_2 as *server* and adopt a star communication setup (letting $\lceil \frac{P_1 N_2}{N_2 - 1} \rceil \leq \lceil \frac{P_2 N_1}{N_1 - 1} \rceil$). Note that, if E_1 obtains the joint feasible set $\mathcal{P}$ first using [10], and then finds $\mathcal{P}^*$ by comparing the function values, the associated download cost is $D_{PSI} = D_{PSI}(P_1, N_2) = \lceil \frac{P_1 N_2}{N_2 - 1} \rceil$.

In contrast, in our achievable scheme to seek for the optimal solution set, E_1 sequentially checks the presence of the set of elements that achieve the best function value. The answers received in the current iteration, after being decoded, decide the set of queries for the next iteration. We present an example before formally describing our scheme.

Example 1 *Consider the following scenario. A universal movie-rating system provides a score between* 1 *to* 5 *to the movies, with* 5 *being the best. Let* $\mathcal{P}_{alph} = \{A, B, C, D, E, F, G, H\}$ *be the collection of movie IDs. Entities* E_1 *and* E_2 *have subsets of these IDs as their feasible sets as*

$$\mathcal{P}_1 = \{A, C, D, G\}, \qquad \mathcal{P}_2 = \{B, C, D, G, H\}, \tag{16}$$

and therefore the incidence vectors as

$$X_1 = [1\ 0\ 1\ 1\ 0\ 0\ 1\ 0]^T, \qquad X_2 = [0\ 1\ 1\ 1\ 0\ 0\ 1\ 1]^T. \tag{17}$$

The goal of the entities is to find the set of movies with the highest score present in the sets of both entities. Let the scores be given by

$$\begin{aligned} &f(B) = f(E) = f(H) = 5, \\ &f(A) = f(C) = f(G) = 4, \\ &f(D) = 3, \\ &f(F) = 2. \end{aligned} \tag{18}$$

Therefore, we have $\boldsymbol{\alpha} = [3, 1]$. *For this problem,* $\mathcal{P}^*$ *is* $\{C, G\}$.

First, E_1 *checks for the presence of* $\{A\}$, $\{C\}$ *or* $\{G\}$ *(the elements that attain the maximum score of* 4 *in* $\mathcal{P}_1$*) in* $\mathcal{P}_2$, *i.e., for* $\mathcal{J}_1 = \{1, 3, 7\}$. *To do so,* E_1 *picks a random vector* h_1 *from* $\mathbb{F}_q^8$ *with* $q > 3$ *and prime, and sends the queries,*

$$Q_1 = h_1, \qquad Q_2 = h_1 + e_1 + e_3 + e_7, \tag{19}$$

to the first two databases of E_2. *We assume that the databases of* E_2 *share a common random*

N_2	database 1	database 2	database 3	database 4
2	h_2, h_3	h_2+e_1, h_3+e_3	-	-
3	h_2	h_2+e_3	h_1+e_1	-
4			h_1+e_1	h_1+e_3

Table 1: Queries sent to check the presence of $\{A, C\}$ in $\mathcal{P}_2$.

variable $S_1 \in \mathbb{F}_q^8$. *The answers returned by the corresponding databases are,*

$$A_1 = Q_1^T X_2 + S_1, \qquad A_2 = Q_2^T X_2 + S_1. \tag{20}$$

Then, entity E_1 *computes* $A_2 - A_1 = 2$ *and concludes that there are* 2 *elements with score* 4 *in* $\mathcal{P}_1 \cap \mathcal{P}_2$. *In the first phase,* E_1 *downloads* 2 *symbols, one from each database.*

Now, to find which elements out of $\{A, C, G\}$ *form the solution set,* E_1 *checks for the presence of any two* $\{A, C\}$ *(and* $\{C, G\}$ *and* $\{A, G\}$*) in* $\mathcal{P}_2$. *Based on the number of databases* N_2 *at* E_2*'s side, the queries will differ for this phase as tabulated in Table 1. For instance, in the case of* $N_2 = 3$*, the answers downloaded by* E_1 *are* $(h_1 + e_1)^T X_2 + S_1$ *from database 3,* $h_2^T X_2 + S_2$ *from database 1, and* $(h_2 + e_3)^T X_2 + S_2$ *from database 2, where the databases share* $S_1, S_2 \in \mathbb{F}_q^8$ *as common random variables. In this phase, the number of downloaded symbols is* 4, 3 *and* 2 *if* N_2 *is* 2, 3 *or* 4*, respectively. If there are more than* 4 *databases then queries in the last row are sent to any two of databases 3 to* N_2.

Now, with the same feasible sets consider a different score mapping where $f(C) = f(G) = 3$ *and the rest of the scores are the same as before, i.e.,*

$$\begin{aligned} &f(B) = f(E) = f(H) = 5, \\ &f(A) = 4, \\ &f(C) = f(D) = f(G) = 3, \\ &f(F) = 2. \end{aligned} \tag{21}$$

Therefore, we have $\boldsymbol{\alpha} = [1, 3]$. *Entity* E_1 *checks for the existence of* $\{A\}$ *in* E_2 *by sending queries* h_1 *and* $h_1 + e_1$ *to two databases of* E_2. *Since the answers from* E_2 *reveal the absence of* A *in* $\mathcal{P}_2$, E_1 *moves to the next-best value* 3 *and checks for the existence of* $\{C\}$, $\{D\}$ *or* $\{G\}$. *On learning that all three elements are common in* $\mathcal{P}_2$, E_1 *directly learns that the optimal solution is* $\{C, D, G\}$. *The download cost in this case is* 4 *if* $N_2 = 2$ *and* 2 *if* $N_2 > 3$.

Next, consider yet another score mapping where $f(G) = 5$, *and* $f(x) = 4$, *otherwise, i.e.,*

$$\begin{aligned} &f(G) = 5, \\ &f(A) = f(B) = f(C) = f(D) = f(E) = f(F) = f(H) = 4. \end{aligned} \tag{22}$$

Therefore, we have $\boldsymbol{\alpha} = [1, 3]$. *This time,* E_1 *starts by checking the existence of* G *by sending queries* h_1 *and* $h_1 + e_7$ *to two databases with* E_2. *By downloading* 2 *symbols as answer,* E_1

finds the optimal solution to be $\{G\}$.

4.1 Proposed Scheme

As illustrated in the example, our scheme consists of two primitives: 1) CarPSI and 2) FindPSI. Let $M_{i_r} := X_{\mathcal{J}_r}^T X_2$ denote the multiplicity of elements with function value f_{i_r} in $\mathcal{P}_1 \cap \mathcal{P}_2$. Before the start of communication, both entities agree on a finite field $\mathbb{F}_q$ with a prime q which satisfies $q > \max_{r\in[L]} \alpha_r$. Now, we describe the functionalities CarPSI and FindPSI for any $r \in [L]$.

4.1.1 CarPSI($X_{\mathcal{J}_r}, N_2$)

CarPSI($X_{\mathcal{J}_r}, N_2$) computes the cardinality M_{i_r} for which E_1 evaluates the dot product of $X_{\mathcal{J}_r}$ with X_2 without revealing $\mathcal{J}_r$ to any database of E_2. Further, CarPSI should not reveal any additional information on X_2 beyond M_{i_r}. Depending on r, there can be two cases.

Case i): If $r = (k-1)(N_2-1)+1$ for some $k \in \mathbb{N}$, E_1 uniformly selects K elements $h_k(j), j \in [K]$ independently from $\mathbb{F}_q$ to form the vector $h_k = [h_k(1), h_k(2), \ldots, h_k(K)]^T$. Now, E_1 randomly selects 2 databases (say the first 2) from the N_2 databases available to E_2 to participate in the protocol. The queries sent are

$$Q_1 = h_k, \qquad Q_2 = h_k + X_{\mathcal{J}_r}. \tag{23}$$

To generate answers, the databases calculate the dot product $Q_n^T X_2$, $n = 1, 2$ and add a new symbol to $S_k \in \mathcal{S}$,

$$A_1 = Q_1^T X_2 + S_k = \sum_{j=1}^{K} h_k(j) X_2(j) + S_k, \tag{24}$$

$$A_2 = Q_2^T X_2 + S_k = \sum_{j=1}^{K} \big(h_k(j) X_2(j) + X_{\mathcal{J}_r}(j) X_2(j)\big) + S_k. \tag{25}$$

All operations are performed in $\mathbb{F}_q$.

Case ii): If $r = (k-1)(N_2-1)+l$, where $l \in \{2, \ldots, N_2-1\}$ for some $k \in \mathbb{N}$, the same vector h_k used in CarPSI($X_{\mathcal{J}_{r-1}}, N_2$) is reused to form the query $h_k + X_{\mathcal{J}_r}$. It is sent to one of the databases n for which h_k was not involved in the query. In response, database n sends $A_n = Q_n^T X_2 + S_k$. Having received the answers, E_1 computes M_{i_r} as,

$$M_{i_r} = A_n - A_1 \tag{26}$$

$$= (Q_n - Q_1)^T X_2 = X_{\mathcal{J}_r}^T X_2. \tag{27}$$

4.1.2 FindPSI($\mathcal{J}_r, M_{i_r}, N_2$)

This functionality evaluates the private set intersection (PSI) between two entities, with the knowledge of the size of the intersection. Since M_{i_r} is known, it suffices if E_1 leaves any single index (say, j) from $\mathcal{J}_r$ out, to learn the set intersection. Let $\bar{\mathcal{J}} = \mathcal{J}_r \setminus \{j\}$ be the set on which PSI is to be executed, with $|\bar{\mathcal{J}}| = \alpha_r - 1$. Let $r = (k-1)(N_2-1) + l$ for some $k \in \mathbb{N}$ and $l \in [0 : N_2 - 2]$, and $\alpha_r - 1 = (N_2 - l - 1) + p(N_2 - 1) + s$, for some $p \in \mathbb{N} \cup \{0\}, s \in [0 : N_2 - 2]$. Depending on l, there are two cases.

Case i) If $l = 0$, then $\alpha_r - 1 = (p+1)(N_2 - 1) + s$. E_1 generates $p+2$ independent, uniformly random vectors $h_{k+1}, \ldots, h_{k+p+2} \in \mathbb{F}_q^K$.

Case ii) If $l \in [N-2]$, $l+1$ databases have been previously sent queries to, using the same random vector h_k in CarPSI. Therefore, E_1 reuses h_k to submit queries to the unused (assume, $l+2, \ldots, N_2$) databases, and picks $p+1$ new random, independent vectors $h_{k+1}, \ldots, h_{k+p+1}$ for the subsequent queries.

E_1 sends $Q_1 = \{h_y,\ y \in [k+1 : k+p+2]\}$ as query in case i) and $Q_1 = \{h_y,\ y \in [k+1 : k+p+1]\}$ as query in case ii) to database 1. To databases $n \in [2 : N_2]$, the queries for the corresponding y are $Q_n = h_y + e_j, j \in \bar{\mathcal{J}}$. In both cases, the answer returned corresponding to each y is

$$A_n = Q_n^T X_2 + S_y, \quad n \in [N_2], \tag{28}$$

where $S_y \in \mathcal{S}$. Formally, the integer y ranges across $[k+1 : k+p+2]$ if $n \in [s+1]$ and across $[k+1 : k+p+1]$ otherwise, in case i). In case ii) if $s \leq l$, then $y \in [k+1 : k+p+1]$ if $n \in [s+1]$, $y \in [k+1 : k+p]$ if $n \in s+2 : l+1$ and $y \in [k : k+p]$ if $n \in [l+2 : N_2]$. The sub-case where $s > l$ can be similarly handled. To decode each element in $X_2(\bar{\mathcal{J}})$, E_1 evaluates $A_n - A_1$. By doing so, E_1 learns $X_2(\mathcal{J}_r)$ since M_{i_r} is known.

4.1.3 Main Algorithm

The complete flow of the achievable scheme is given in Algorithm 1. First, E_1 creates the sets of indices $\mathcal{J}_r, r \in [L]$ and the vector $\boldsymbol{\alpha}$ of length L. Next, it executes CarPSI($X_{\mathcal{J}_1}, N_2$) to find M_{i_1} for the best function value f_{i_1}. Based on M_{i_1} and α_1, there are 3 cases:

i) If $M_{i_1} = 0$, E_2 does not possess any element in $\mathcal{P}_2$ in common with E_1 that yields the same function value f_{i_1}. E_1 proceeds to the next best function value f_{i_2} and finds M_{i_2} by executing CarPSI($X_{\mathcal{J}_2}, N_2$).

ii) If $M_{i_1} = \alpha_1$, then f_{i_1} is a feasible function value with $\mathcal{P}_{alph}(\mathcal{J}_1)$ as the solution set.

iii) If $1 \leq M_{i_1} \leq \alpha_1 - 1$, E_1 proceeds to find the solution set which consists of M_{i_1} out of α_1 indices in $\mathcal{J}_1$ by implementing FindPSI($\mathcal{J}_1, M_{i_1}, N_2$).

Algorithm 1 Scheme to find $\mathcal{P}^*$

Input: $\mathcal{P}_1, f, \boldsymbol{\alpha}, N_2$
Output: $\mathcal{P}^*$

$L \leftarrow$ length of $\boldsymbol{\alpha}$
for $r \in \{1, 2, \ldots, L\}$ **do**
  if $r < L$ or $\alpha_L > 1$ **then**
    $M_{i_r} = \text{CarPSI}(X_{\mathcal{J}_r}, N_2)$
    if $1 \leq M_{i_r} \leq \alpha_r - 1$ **then**
      $\mathcal{P}^* = \text{FindPSI}(\mathcal{J}_r, M_{i_r}, N_2)$
        break
    else if $M_{i_r} == \alpha_r$ **then**
      $\mathcal{P}^* = \mathcal{P}_{alph}(\mathcal{J}_r)$
        break
  else
    $\mathcal{P}^* = \mathcal{P}_{alph}(\mathcal{J}_r)$
return $\mathcal{P}^*$

In general, CarPSI is executed according to i) till the returned cardinality $M_{i_r} \geq 1$, $r \in [L]$. If $\alpha_r = M_{i_r}$, E_1 directly sets $\mathcal{P}^* = \mathcal{P}_{alph}(\mathcal{J}_r)$. Otherwise, E_1 executes FindPSI. If the search continues and $M_{i_r} = 0$ for every $r \leq L-1$, the solution is contained in $\mathcal{P}_{alph}(\mathcal{J}_L)$, since we assumed that $|\mathcal{P}_1 \cap \mathcal{P}_2| \geq 1$.

Remark 1 *One can view the aforementioned algorithm as a successive implementation of a ThPSI scheme with threshold $t = 1$, on disjoint subsets of $\mathcal{P}_1$ corresponding to $\mathcal{J}_r$, $r \in [L]$ till the threshold is met on the intersection cardinality. This is done by computing $M = X_1^T X_2$ with CarPSI(X_1, N_2) on $\mathbb{F}_q$ having $q > \min(P_1, P_2)$, prime, and FindPSI(X_1, M, N_2) if $t \leq M < P_1$.*

4.2 Proof of Privacy

Each individual database of E_2 observes an independent, uniformly random vector over $\mathbb{F}_q^K$, in all rounds of CarPSI and FindPSI. This preserves E_1's privacy (6).

For E_2's privacy (5), we guarantee that E_1 learns no more than $\mathcal{I}_{nom,2}$. From the answers and queries, E_1 learns that the elements in $\mathcal{P}_1$ corresponding to indices $\cup_{r=1}^{R-1} \mathcal{J}_r$ are all absent in E_2's constraint set, $\mathcal{P}_2$, i.e., X_2 is 0 at those indices, and the value of X_2 at indices $\mathcal{J}_R$. Thus, E_1 learns X_2 at $\cup_{r=1}^{R} \mathcal{J}_r$ which is a subset of the support of X_1 and

$$I(\mathcal{P}_2; Q_{1:N_2}, A_{1:N_2}, \mathcal{P}_1, f) = H(\mathcal{P}_2) - H(\mathcal{P}_2 | Q_{1:N_2}, A_{1:N_2}, \mathcal{P}_1, f) \tag{29}$$

$$= H(X_2) - H\big(X_2([K] \setminus \cup_{r=1}^{R} \mathcal{J}_r) \mid X_2(\cup_{r=1}^{R} \mathcal{J}_r)\big) \tag{30}$$

$$= H\big(X_2(\cup_{r=1}^{R} \mathcal{J}_r)\big), \tag{31}$$

where we used the fact that X_i is a sufficient statistic for $\mathcal{P}_i$.

4.3 Communication Cost

R is the least value of $r \in [L]$ in Algorithm 1 at which $M_{i_r} \geq 1$. The download cost for the sequential CarPSI is equal to $D_{PSI}(R, N_2)$ which is $\lceil \frac{RN_2}{N_2-1} \rceil$. This is the download cost if $M_{i_R} = \alpha_R$.

Otherwise, let $R = k(N_2 - 1) + l$, for some $k \in \mathbb{N} \cup \{0\}$. If $l = 0$, the download cost in FindPSI is $\lceil \frac{(\alpha_R - 1)N_2}{N_2-1} \rceil$ and that from CarPSI is $\frac{RN_2}{N_2-1}$. Their sum gives (13). With $l \in [N_2 - 2]$, after CarPSI$(X_{\mathcal{J}_R}, N_2)$, $l + 1$ out of N_2 databases are used. The remaining $N_2 - (l + 1)$ databases continue to send answers for FindPSI using the same random vector for query and common randomness symbols, respectively, resulting in the partial download cost of

$$\left\lceil \frac{RN_2}{N_2 - 1} \right\rceil + N_2 - l - 1 = (k + 1)N_2. \tag{32}$$

This is followed by downloads for $\alpha_R - 1 - (N_2 - l - 1)$ indices which costs $D_{PSI}(\alpha_R - N_2 + l, N_2)$, which when added to (32), yields

$$D = (k + 1)N_2 + \left\lceil \frac{(\alpha_R - N_2 + l)N_2}{N_2 - 1} \right\rceil \tag{33}$$

$$= \left\lceil \frac{(\alpha_R - 1)N_2 + (k(N_2 - 1) + l)N_2}{N_2 - 1} \right\rceil \tag{34}$$

$$= \left\lceil \frac{(\alpha_R - 1 + R)N_2}{N_2 - 1} \right\rceil. \tag{35}$$

There are D answers downloaded, each of which is a response to a query of K symbols, hence, the upload cost $U = KD$ and the communication cost $C = (K + 1)D$.

4.4 Common Randomness Cost

In the scheme, E_1 requires $\lceil \frac{R+\alpha_R-1}{N_2-1} \rceil$ random vectors for generating queries. This corresponds to equal amount of common randomness symbols $\mathcal{S}$ shared between the databases of E_2. However, since R and α_R are not known beforehand, the databases of E_2 share a set of common randomness symbols $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ where $m = \lceil \frac{P_1}{N_2-1} \rceil$, independent of f, $\mathcal{P}_1$ and $\mathcal{P}_2$. The databases do not collude hereafter.

4.5 Comparison with the Naive Scheme

We now compare our scheme with a scheme where E_1 first finds the joint feasible set $\mathcal{P} = \mathcal{P}_1 \cap \mathcal{P}_2$ privately, following the existing PSI protocol in [10], and then evaluates f at the values in $\mathcal{P}$ only, selects the best ones among them to obtain $\mathcal{P}^*$. Finally, E_1 conveys this directly to E_2. We refer to this as the *naive scheme*.

Here, the entire $\mathcal{P}$ is leaked to E_1. Thus, the values of X_2 at indices in $\mathcal{J}_r$ $\forall r \in [L]$ are revealed to E_1 which is the support set of X_1. Thus,

$$I(\mathcal{P}; \mathcal{P}_2 | \mathcal{P}_1, f) = H(X_2(\cup_{r=1}^{L} \mathcal{J}_r)) \geq H(X_2(\cup_{r=1}^{R} \mathcal{J}_r)) = \mathcal{I}_{nom,2}. \tag{36}$$

Equality holds only if $R = L$, which is when $M_{i_r} = 0$ for all $r \in [L-1]$.

Since the upload cost is K times the download cost in both schemes, it suffices to compare the communication efficiency in terms of the download cost. On one hand, the naive scheme requires $\lceil \frac{P_1 N_2}{N_2 - 1} \rceil$ downloaded symbols of $\mathbb{F}_q$. For Example 1, the download cost in the naive approach is $\lceil \frac{5N_2}{N_2-1} \rceil$, irrespective of f.

Remark 2 *Alternative to our proposed scheme, applying the scheme in [10] sequentially over the elements in $\mathcal{P}_1$ corresponding to indices $\mathcal{J}_1$, $\mathcal{J}_2$ till $\mathcal{J}_R$ also reveals $\mathcal{I}_{nom,2}$ to E_1. The download cost incurred for this scheme is $\lceil \frac{(\sum_{r=1}^{R} \alpha_r) N_2}{N_2 - 1} \rceil$. Clearly, it is less communication-efficient due to higher download cost compared to D, except when $\alpha_r = 1$ for all $r \in [R-1]$, in which case, they are equal.*

Remark 3 *Under uniform random mapping of f, with a fixed realization of $\mathcal{P}_1$ and $\mathcal{P}_2$, the probability P_{eq} that the download costs D and D_{PSI} are equal, is small. To support this, we provide the following calculation, and a numerical result.*

With a fixed realization of $\mathcal{P}_1$ and $\mathcal{P}_2$, the random variable $R \in [L]$ and vector $\boldsymbol{\alpha}$ depend on the realization of f. The maximum value of L (hence R) is $R_{\max} = \min(\tau, P_1 - M + 1)$. Further, for any $R \in [L]$, $M_{i_R} \leq \alpha_R$. If $M_{i_R} = \alpha_R$, we drop the FindPSI phase, and the download cost is $\lceil \frac{RN_2}{N_2 - 1} \rceil$. For this to equal D_{PSI}, we require $R = P_1$, which implies $M = 1$. Hence, $R = L$ and our algorithm skips the last CarPSI step as well, in which case the actual download cost is $\lceil \frac{(L-1)N_2}{N_2 - 1} \rceil$. As a result, $D_{PSI} > D$ when $M_{i_R} = \alpha_R$.

Therefore, M_{i_R} must be strictly less than α_R so that FindPSI is executed. Thus, $D = D_{PSI}$ if $\{R + \alpha_R = P_1 + 1\}$ and $\{M_{i_R} < \alpha_R\}$ are simultaneously satisfied. Note that, for such events $\alpha_r = 1$ for all $r < R$. Further, $M_r = 0$ for all $r < R$, hence the function value of every element in $\mathcal{P} = \mathcal{P}_1 \cap \mathcal{P}_2$ is f_{i_R}, i.e., optimum. With $M \geq 1$, the corresponding probability P_{eq} is,

$$\begin{aligned}
P_{eq} &= \mathbb{P}(R + \alpha_R = P_1 + 1, M_{i_R} < \alpha_R) && (37)\\
&= \sum_{r=1}^{R_{\max}} \mathbb{P}(R = r, \alpha_r = P_1 + 1 - r, M_{i_R} < P_1 + 1 - r) && (38)\\
&= \sum_{r=1}^{R_{\max}} \mathbb{P}(R = r, \alpha_r = P_1 + 1 - r) \mathbb{P}(M < P_1 + 1 - r) && (39)\\
&= \sum_{r=1}^{R_{\max}} P_{eq}(r) \times \mathbb{1}_{\{M < P_1 - r + 1\}}, && (40)
\end{aligned}$$

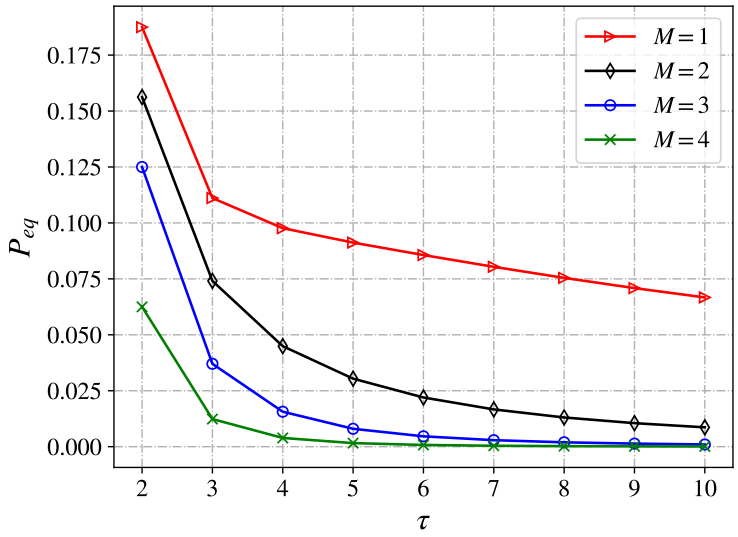


Figure 3: P_{eq} against τ for different values of M with $N = 2$.

where (38) *follows from* (37) *using* $\mathbb{P}(M_{i_R} < P_1 + 1 - r | R = r, \alpha_r = P_1 + 1 - r) = \mathbb{P}(M < P_1 + 1 - r)$ *since* $\mathcal{P} = \mathcal{P}^*$. *Further, since* M *is determined by* $\mathcal{P}_1$ *and* $\mathcal{P}_2$, $\mathbb{P}(M < P_1 + 1 - r)$ *is an indicator as in* (40). *If* $r = 1$, *we have*

$$P_{eq}(1) = \tau \left(\frac{1}{\tau}\right)^{P_1}, \tag{41}$$

since τ *out of* τ^{P_1} *function realizations assume equal function value over* $\mathcal{P}_1$. *For* $1 < r \leq R_{\max}$,

$$P_{eq}(r) = \binom{P_1 - M}{r - 1} (r-1)! \left[\sum_{j=r-1}^{\tau-1} \binom{j-1}{r-2} \times (\tau - j) \right] \times \left(\frac{1}{\tau}\right)^{P_1}. \tag{42}$$

The term preceding the summation in (42) *follows by sampling* $f_{i_1}, f_{i_2}, \ldots, f_{i_{r-1}}$ *with ordering from the* $P_1 - M$ *elements in* $\mathcal{P}_1$ *absent in* $\mathcal{P}$. *Fixing* $f_{i_{r-1}} = j$, *we choose* $f_{i_1}, \ldots, f_{i_{r-2}}$ *from* 1 *to* $j - 1$, *as better function values preceding* $f_{i_{r-1}}$, *in* $\binom{j-1}{r-2}$ *ways. The remaining* $\alpha_R = P_1 - r + 1$ *elements take the same function value* f_{i_R}, *which has* $\tau - j$ *choices in the set* $\{j+1, j+2, \ldots, \tau\}$ *and we sum over all* $j \in \{r-1, \ldots, \tau-1\}$. *This expression, multiplied by the uniform probability of function realization, yields* (42).

To show an example that (40) *is small, in Fig. 3, we plot the probability* P_{eq} *with respect to the size of function range* τ *for various values of* M. *We follow the example of movie scores in Section 1 and fix* $P_1 = 5$ *while the scores are allotted from the set* $\mathcal{T} = [\tau]$ *with* $\tau \in \{2, 3, \ldots, 10\}$. *For a fixed* τ, *increasing* M *from* 1 *to* 4 *decreases the value of* P_{eq}.

5 The Case of $N > 2$ Entities: Ring Setup

The underlying idea of the achievable scheme, similar to the case of $N = 2$, is to apply ThPSI with $t = 1$ on each set of equi-cost elements in $\mathcal{P}_{alph}$ sequentially, till the threshold condition of non-empty intersection is satisfied. We present a PSI scheme under the ring setup, and show that the communication cost of our DOFSP scheme is less than or equal to that of the PSI scheme. Through numerical results, we show that the probability that the communication costs of the two schemes are equal, is low.

We begin the description our scheme with an example.

Example 2 *Continuing with the movie example, let $\mathcal{P}_{alph} = \{A, B, C, D, E, F, G, H, I, J\}$ be the set of movies and f be their scores assigned as follows:*

$$f(x) = \begin{cases} 1, & x \in \{A\}, \\ 2, & x \in \{B, E\}, \\ 3, & x \in \{F, G, H\}, \\ 4, & x \in \{I, J\}, \\ 5, & x \in \{C, D\}. \end{cases} \tag{43}$$

Then, with $N = 4$ entities, let the individual feasible sets be as follows

$$\begin{aligned} \mathcal{P}_1 &= \{A, C, D, G, J\} \quad \implies X_1 = [1,0,1,1,0,0,1,0,0,1]^T, \hat{X}_1 = [1,1,0,1,0,1,0,0,0,1]^T \\ \mathcal{P}_2 &= \{B, E, G, H, I, J\} \quad \implies X_2 = [0,1,1,0,1,0,1,1,1,1]^T, \hat{X}_2 = [0,0,1,1,0,1,1,1,1,0]^T \\ \mathcal{P}_3 &= \{A, B, C, F, G, I\} \quad \implies X_3 = [1,1,1,0,0,1,1,0,1,0]^T, \hat{X}_3 = [1,0,1,0,1,1,0,1,0,1]^T \\ \mathcal{P}_4 &= \{C, D, G, H, I\} \quad \implies X_4 = [0,0,1,1,0,0,1,1,1,0]^T, \hat{X}_4 = [1,1,1,0,0,1,1,0,0,0]^T \end{aligned} \tag{44}$$

where $\hat{X}_i$ is X_i after permuting the indices as

$$\hat{X}_i(1,2,3,4,5,6,7,8,9,10) = X_i(3,4,9,10,6,7,8,2,5,1). \tag{45}$$

Let $N_i = 2$ for each E_i, implying $\mathcal{N}_{eff} = \{1, 2\}$. Let $q = 5$, and $E_i, i \in \{2, 3, 4\}$ share uniformly and independently generated randomness vectors $S_i^{(1)} \in \mathbb{F}_q^2, S_i^{(2)} \in \mathbb{F}_q^2, S_i^{(3)} \in \mathbb{F}_q^3, S_i^{(4)} \in \mathbb{F}_q^2$ and $S_i^{(5)} \in \mathbb{F}_q$ among each database pair. First, the entities check for the presence of $\{C, D\}$ (elements that map to f_1) in $\mathcal{P}$. Towards this, E_1 generates h_1 uniformly at random from $\mathbb{F}_q^2$ to send the queries

$$Q_{1,1}^{(1)} = h_1, \quad Q_{1,2}^{(1)} = h_1 + \hat{X}_1(1,2), \tag{46}$$

$$Q_{2,j}^{(1)} = Q_{1,j}^{(1)} \circ \hat{X}_2(1,2) + S_2^{(1)}, \quad Q_{3,j}^{(1)} = Q_{2,j}^{(1)} \circ \hat{X}_3(1,2) + S_3^{(1)}, \quad j \in \{1,2\} \tag{47}$$

$$A_1^{(1)} = {Q_{3,1}^{(1)}}^T \hat{X}_4(1,2) + S_4^{(1)}(2), \quad {Q_{3,2}^{(1)}}^T \hat{X}_4(1,2) + S_4^{(1)}(2), \tag{48}$$

where the superscript indicates the round index, and the symbol $\circ$ *indicates elementwise multiplication. Next,* E_1 *evaluates*

$$\begin{aligned} A_2^{(1)} - A_1^{(1)} =& (((h_1 + \hat{X}_1(1,2)) \circ \hat{X}_2(1,2) + S_2^{(1)}) \circ \hat{X}_3(1,2) + S_3^{(1)})^T \hat{X}_4(1,2) + S_4^{(1)}(2) \\ & - ((h_1 \circ \hat{X}_2(1,2) + S_2^{(1)}) \circ \hat{X}_3(1,2) + S_3^{(1)})^T \hat{X}_4(1,2) + S_4^{(1)}(2) \qquad (49) \\ =& (\hat{X}_1(1,2) \circ \hat{X}_2(1,2) \circ \hat{X}_3(1,2))^T \hat{X}_4(1,2) = 0. \qquad (50) \end{aligned}$$

Since the intersection is empty, no elements mapped to 5 *are present in* $\mathcal{P}$*. Now,* E_1 *proceeds to the next best value* 4 *and generates a new random vector* $h_2 \in \mathbb{F}_q^2$ *and similarly evaluates* $A_2^{(2)} - A_1^{(2)} = (\hat{X}_1(3,4) \circ \hat{X}_2(3,4) \circ \hat{X}_3(3,4))^T \hat{X}_4(3,4)$ *which is also* 0*. Next,* E_1 *generates* $h_3 \in \mathbb{F}_q^3$ *and similarly evaluates* $A_2^{(3)} - A_1^{(3)} = (\hat{X}_1(5,6,7) \circ \hat{X}_2(5,6,7) \circ \hat{X}_3(5,6,7))^T \hat{X}_4(5,6,7)) = 1$*, which is the first non-zero cardinality. Now,* E_1 *sends two elements from* $\mathbb{F}_q$*, one each to the databases of* E_4*, which signals them to send the following answers to* E_1*,*

$$A'_j = (Q_1^{(3)} \circ \hat{X}_4)(5,6) + S_4^{(3)}(1,2), \quad j \in \{1,2\}. \tag{51}$$

Using these, E_1 *calculates* $A'_2 - A'_1 = \hat{X}_1(5,6) \circ \hat{X}_2(5,6) \circ \hat{X}_3(5,6,) \circ \hat{X}_4(5,6) = [0,1]^T$*. This gives* $\mathcal{P}^* = \{G\}$*, since the cardinality is* 1*. The communication protocol stops after* 3 *rounds where each of the first two rounds required* $2 \cdot 3 \cdot 2 + 2 = 14$ *symbols and the last round required* $2 \cdot 3 \cdot 3 + 2 + 2 + 4 = 26$ *symbols, hence* $C_{ring} = 14 + 14 + 26 = 54$*.*

5.1 Scheme Preliminaries

First, we extend the two-entity ($N = 2$) functionalities CarPSI and FindPSI from Section 4 to the ring setup, to develop new primitives CarPSI-ring and FindPSI-ring.[1] In CarPSI-ring, E_1 privately obtains the cardinality M given by,

$$M = \mathbf{1}^T (X_1 \circ X_2 \ldots \circ X_N) = (X_1 \circ X_2 \ldots \circ X_{N-1})^T X_N, \tag{52}$$

without learning the actual intersection $\mathcal{P}$, where $\mathbf{1}$ is the $K \times 1$ all ones vector and $\circ$ denotes the elementwise product operation. FindPSI-ring is executed after this, if the cardinality exceeds a certain value. We assume that each entity has at least two replicated non-colluding databases storing the incidence vectors of the feasible sets. Before the start of CarPSI-ring, the entities agree on an operating field $\mathbb{F}_q$ where $q > \min_{i \in [N]} P_i$ and prime. We let the shared common randomness $\mathcal{S}_i$ be a random vector $S_i \in \mathbb{F}_q^K$ whose entries $S_i(k)$ are picked uniformly at random and independently from $\mathbb{F}_q$ for each $i \in [2:N]$ and $k \in [K]$.

[1] Similar to the $N = 2$ case, these primitives can be executed to attain a ThPSI scheme. After CarPSI-ring, if the resulting cardinality M is t or greater, E_1 executes FindPSI-ring, unless $M = P_1$, which is the trivial case that $\mathcal{P} = \mathcal{P}_1$.

5.1.1 CarPSI-ring(X_1)

First, E_1, using a private random vector h drawn uniformly from $\mathbb{F}_q^K$ generates two queries $Q_{1,1} = h_1$ and $Q_{1,2} = h_1 + X_1 = Q_{1,1} + X_1$ and sends them to databases 1 and 2 of E_2, respectively. On receiving $Q_{1,j}$ database $j = 1, 2$ computes its elementwise product with X_2, and adds the randomness vector S_2 to it. Then, database $j \in \{1, 2\}$ sends the computed value as the query $Q_{2,j} = Q_{1,j} \circ X_2 + S_2$ to database j of E_3, and so on. In general, for each $i \in [N-2]$, database j of E_i uses its incoming query to send the following to database j of E_{i+1},

$$\begin{aligned} Q_{i+1,j} =& Q_{i,j} \circ X_i + S_i && (53)\\ =& (Q_{i-1,j} \circ X_{i-1} + S_{i-1}) \circ X_i + S_i && (54)\\ & \vdots \\ =& ((Q_{1,j} \circ X_2 + S_2) \circ X_3 + S_3) \circ \ldots \circ X_i + S_i. && (55) \end{aligned}$$

Finally, the databases of E_N use their incoming queries $Q_{N-1,j}$ to compute and send the following answers to E_1,

$$A_1 = {Q_{N-1,1}}^T X_N + S_N(K), \quad (56)$$
$$A_2 = {Q_{N-1,2}}^T X_N + S_N(K), \quad (57)$$

where $S_N(K)$ is the Kth coordinate of S_N. From the received answers, E_1 computes $A_2 - A_1$.

To prove that the leader obtains M, we first prove the fact that, for all $i \in [N-1]$,

$$Q_{i,2} = Q_{i,1} + (X_1 \circ \ldots \circ X_i). \quad (58)$$

We show this by induction on i. For $i = 1$, it is trivially true. Assume that this is true for $i = k$ for some $k \in [2 : N-2]$. Then, we have,

$$\begin{aligned} Q_{k+1,2} &= Q_{k,2} \circ X_{k+1} + S_{k+1} && (59)\\ &= (Q_{k,1} + (X_1 \circ \ldots \circ X_k)) X_{k+1} + S_{k+1} && (60)\\ &= Q_{k,1} \circ X_{k+1} + (X_1 \circ \ldots \circ X_{k+1}) + S_{k+1} && (61)\\ &= Q_{k+1,1} + (X_1 \circ \ldots \circ X_{k+1}), && (62) \end{aligned}$$

as required. Therefore,

$$\begin{aligned} A_2 - A_1 &= (Q_{N-1,2} - Q_{N-1,1})^T X_N + S_N(K) - S_N(K) && (63)\\ &= (X_1 \circ X_2 \circ \ldots \circ X_{N-1})^T X_N. && (64) \end{aligned}$$

To prove privacy, note that the queries sent by E_1 to E_2 protect the privacy of X_1 (hence,

of $\mathcal{P}_1$) since,

$$I(Q_{1,1}; X_1|S_2, X_2) = I(h_1; X_1) = 0, \tag{65}$$
$$I(Q_{1,2}; X_1|S_2, X_2) = I(h_1 + X_1; X_1) = 0, \tag{66}$$

where (65) is true because h_1, S_2 and X_2 are independent of X_1, and (66) uses Shannon's one-time padding [22]. For $i \in [2 : N-1]$, the privacy of E_1 through E_i is preserved against an individual database of E_{i+1}. We show this by induction on i. Let $i = 2$, for database 1,

$$I(Q_{2,1}; X_1, X_2|S_3, X_3) = I(h_1 \circ X_2 + S_2; X_1, X_2) \tag{67}$$
$$= I(h_1 \circ X_2 + S_2; X_1) + I(h_1 \circ X_2 + S_2; X_2|X_1) \tag{68}$$
$$= 0, \tag{69}$$

where S_3, X_3 are dropped in (67) since they are independent of $Q_{2,1}, X_1$ and X_2. In (68), the first term is 0 because h_1, X_2 and S_2 are independent of X_1 and the second term is 0 by one-time padding with S_2. Similarly, for database 2,

$$I(Q_{2,2}; X_1, X_2|S_3, X_3) = I((h_1 + X_1) \circ X_2 + S_2; X_1, X_2) \tag{70}$$
$$= I((h_1 + X_1) \circ X_2 + S_2; X_1) + I((h_1 + X_1) \circ X_2 + S_2; X_2|X_1) \tag{71}$$
$$= 0. \tag{72}$$

Let $I(Q_{i-1,j}; X_1, \ldots, X_{i-1}|X_i, S_i) = 0$, $\forall j = 1, 2$, then,

$$I(Q_{i,j}; X_1, \ldots, X_i|X_{i+1}, S_{i+1}) = I(Q_{i-1,j} \circ X_i + S_i; X_1, \ldots, X_{i-1}) + I(Q_{i-1,j} \circ X_i + S_i; X_i|X_1, \ldots, X_{i-1}) \tag{73}$$
$$= 0, \tag{74}$$

where the first term in (73) is 0 by the induction hypothesis and the fact that the conditioning terms, being independent, can be dropped. The second term in (73) is 0 by one-time padding. The answers A_1 and A_2 are one-time padded with the Kth entry of S_N, hence, they appear to be chosen uniformly at random from $\mathbb{F}_q$. The evaluation $A_2 - A_1$ reveals only M to E_1 and maintains the privacy of $\mathcal{P}$. The communication cost for CarPSI-ring is $2(N-1)K + 2$ symbols.

Remark 4 *When $N = 2$, CarPSI-ring reduces to CarPSI(X_1, N_2) as described in Section 4.1.1 with $M = X_1^T X_2$.*

5.1.2 FindPSI-ring(M)

Here, E_1 sends back two random symbols of $\mathbb{F}_q$, one to each of the responding databases (say 1 and 2) of E_N to signal them for answers. This time, the databases of E_N reuse the

received queries from CarPSI-ring, and respond with,

$$A'_1 = (Q_{N-1,1} \circ X_N)(1 : K-1) + S_N(1 : K-1), \tag{75}$$

$$A'_2 = (Q_{N-1,2} \circ X_N)(1 : K-1) + S_N(1 : K-1). \tag{76}$$

Thus, the answers are $(K-1)$-length vectors since the last entries of $Q_{N-1,j}$ and X_N are not included. We denote the incidence vector of $\mathcal{P}$ by $X_\cap$, i.e., $X_\cap = X_1 \circ X_2 \circ \ldots \circ X_N$. E_1 directly obtains $X_\cap(1 : K-1)$ from $A'_2 - A'_1$ due to (58). This completes the retrieval of $X_\cap$, and E_1 learns the intersection $\mathcal{P}$, since $M = \mathbf{1}^T X_\cap$ is already known to it.

To show privacy, A'_1 and A'_2 individually hide X_N because each of them is one-time padded with $S_N(1 : K-1)$, which is independent of $S_N(K)$. After computing $A'_2 - A'_1$, E_1 only learns the function $X_2 \circ \ldots \circ X_N$ and nothing beyond on $X_2, \ldots, X_N$ because of (66). Further, the databases of E_N receive random symbols from E_1, which reveal no more information than the threshold condition being satisfied. The communication cost is $2+2(K-1) = 2K$, where the first term is contributed by the symbols sent by E_1 and the second term arises from the answers of E_N.

With these primitives at hand, we proceed with the steps of our DOFSP scheme.

5.2 Scheme Description

Let f_R be the optimum function value for the given objective. Then, to retrieve $\mathcal{P}^*$, E_1 executes the CarPSI-ring primitive over partitions of X_i till the Rth round and one FindPSI-ring primitive. We denote the query $Q_{i,j}$ as the collection of all queries $Q_{i,j}^{(r)}, \forall r \in [R]$ sent over R rounds. Similarly, $A_j = (\cup_{r=1}^{R} A_j^{(r)}) \cup A'_j$. The operating field $\mathbb{F}_q$ is chosen such that q is a prime greater than $\max_{r\in[T]} \mu_r$. This is to ensure correct evaluation of the set intersection cardinalities in the CarPSI-ring primitive.

Partitioning X_i into Sub-Alphabets: Each entity E_i groups X_i into T partitions, where each partition comprises the equi-cost elements. For the objective function f, the entities use a fixed permutation $\Pi_f : [K] \to [K]$, X_i is rearranged into $\hat{X}_i := \Pi_f(X_i)$ for each $i \in [N]$. For each partition $r \in [T]$, let $\mathcal{I}_r := (k_1, k_2, \ldots, k_{\mu_r})$, with $k_1 < k_2 < \ldots < k_{\mu_r}$ denote an ordered set such that $f(\mathcal{P}_{alph}(k_1)) = f(\mathcal{P}_{alph}(k_2)) = \ldots = f(\mathcal{P}_{alph}(k_{\mu_r})) = f_r$, which is the rth best function value. Then,

$$\Pi_f(k_j) = \sum_{v=1}^{r-1} \mu_v + j \implies \hat{X}_i\Big(\sum_{v=1}^{r-1} \mu_v + j\Big) = X_i(k_j), \qquad \forall j \in [\mu_r]. \tag{77}$$

Note that, the permutation Π_f is known to all entities since it is derived directly from f.

Let, for each $i \in [N]$, $X(\mathcal{I}_r)_{\mu_r \times 1}$ denote the column vector with entries at $\mathcal{I}_r$, then

$$\hat{X}_i = \begin{bmatrix} X_i(\mathcal{I}_1)^T & X_i(\mathcal{I}_2)^T \dots & X_i(\mathcal{I}_T)^T \end{bmatrix}^T, \tag{78}$$

is stored across the databases in $\mathcal{N}_{eff}$ databases for each E_i.

Assignment of Common Randomness: The databases in $\mathcal{N}_{eff}$ of non-leader entities generate and share common randomness vectors as follows.

1. If $N_{eff} = 2$, databases 1 and 2 of E_2 through E_N generate randomness vectors uniformly at random from $\mathbb{F}_q^{\mu_r}$ for each $r \in [T]$, where r denotes the round of communication. The vectors are denoted by $S_i^{(r)}$, $i \in [2:N]$.

2. Next, if $N_{eff} > 2$, the databases $\{j, j+1\} \in \mathcal{N}_{eff}$ of E_i are assigned the rounds $r \in \mathcal{R}_j$ in a cyclic fashion, modulo N_{eff}, corresponding to which they store $S_i^{(r)} \in \mathbb{F}_q^{\mu_{r_j}}$. Specifically, for each $i \in [2:N]$:

   - If N_{eff} is even,

   $$\mathcal{R}_j = \left\{ kN_{eff}/2 + \lceil \frac{j}{2} \rceil,\ k \in \left[0 : \left\lfloor \frac{2(T - \lceil \frac{j}{2} \rceil)}{N_{eff}} \right\rfloor \right] \right\}. \tag{79}$$

   - If N_{eff} is odd,

   $$\mathcal{R}_j = \begin{cases} \left\{ \lfloor kN_{eff}/2 \rfloor + \frac{j+1}{2},\ k \in \left[0 : \lceil \frac{2T-j-1}{N_{eff}} \rceil \right] \right\}, & j \text{ is odd}, \\ \left\{ \lceil kN_{eff}/2 \rceil + \frac{j}{2},\ k \in \left[0 : \lfloor \frac{2T-j}{N_{eff}} \rfloor \right] \right\}, & j \text{ is even}. \end{cases} \tag{80}$$

   For example, if $T = 5$ and $N_{eff} = 4$, $\mathcal{R}_1 = \mathcal{R}_2 = \{1, 3, 5\}$ and $\mathcal{R}_3 = \mathcal{R}_4 = \{2, 4\}$. On the other hand, if $N_{eff} = 3$, $\mathcal{R}_1 = \{1, 2, 4, 5\}$, $\mathcal{R}_2 = \{1, 3, 4\}$ and $\mathcal{R}_3 = \{2, 3, 5\}$.

Query and Answer Generation: If $N_{eff} > 2$, database j of E_2 through E_N send queries and answers for rounds in $\mathcal{R}_j$ as given by (79) and (80). If $N_{eff} = 2$, then each database participates in all the rounds.

Next, we provide a description of our scheme for $N_{eff} = 2$ and that for $N_{eff} > 2$ can be similarly derived, taking care that, a distinct random vector is used, each time the same database is queried. The quantity N_{eff} does not affect the overall communication, computation or randomness costs, and only affects the load on individual databases. That is, a higher value of N_{eff} reduces the load of computation, since each database receives queries for fewer number of rounds.

In the first round, E_1 executes CarPSI-ring on $\hat{X}_1([\mu_1])$. That is, it generates a vector h_1 uniformly at random from $\mathbb{F}_q^{\mu_1}$ and sends $Q_{1,1}^{(1)} = h_1$ and $Q_{1,2}^{(1)} = h_1 + \hat{X}_1([\mu_1])$ to databases

$\{1,2\}$ of E_2. Next, database j of E_2 sends $Q_{2,j}^{(1)} = Q_{1,j}^{(1)} \circ \hat{X}_2([\mu_1])$ to database j of E_3. This continues for $i \in [2:N-1]$ following which E_N sends answers to E_1 as follows

$$A_j^{(1)} = Q_{N-1,1}^{(1)T} \hat{X}_N([\mu_1]) + S_N^{(1)}(\mu_1), \quad j = 1,2. \tag{81}$$

Using these answers, E_1 evaluates $\hat{X}_\cap(\mu_1) = A_2^{(1)} - A_1^{(1)}$ to retrieve M_1, i.e., the cardinality of elements in $\mathcal{P}$ that take the value f_1. If this is 0, E_1 executes the next round of CarPSI-ring.

In round $r \geq 2$, E_1 generates a fresh random vector $h_r \in \mathbb{F}_q^{\mu_r}$ to send the queries $Q_{1,1}^{(r)} = h_r$ and $Q_{1,1}^{(r)} = h_r + \hat{X}_1\big(\big[\sum_{v=1}^{r-1}\mu_v + 1 : \sum_{v=1}^{r-1}\mu_v + \mu_r\big]\big)$. The same set of operations are followed by the subsequent entities and, for $i \in [N-1]$, $j \in \mathcal{N}_{eff}$,

$$Q_{i+1,j}^{(r)} = Q_{i,j}^{(r)} \circ \hat{X}_i\big(\big[\sum_{v=1}^{r-1}\mu_v + 1 : \sum_{v=1}^{r-1}\mu_v + \mu_r\big]\big) + S_i^{(r)}(\mu_r). \tag{82}$$

Further, the answer returned by database $j \in \mathcal{N}_{eff}$ of E_N is,

$$A_j^{(r)} = Q_{N-1,j}^{(r)T} \hat{X}_N\big(\big[\sum_{v=1}^{r-1}\mu_v + 1 : \sum_{v=1}^{r-1}\mu_v + \mu_r\big]\big) + S_N^{(r)}(\mu_r). \tag{83}$$

Let $r = R$ denote the round for which $M_r \geq 1$ for the first time. Now, E_1 executes FindPSI-ring(M_R), receiving the answers,

$$A'_j = (Q_{N-1,j}^{(R)} \circ \hat{X}_N)([\sum_{v=1}^{R-1}\mu_v + 1 : \sum_{v=1}^{R-1}\mu_v + \mu_R - 1\big]) + S_N^{(R)}([\mu_R - 1]), \quad j = 1,2. \tag{84}$$

Next, E_1 evaluates $A'_2 - A'_1$ to obtain $\hat{X}_1 \circ \ldots \circ \hat{X}_N$ at indices $[\sum_{v=1}^{R-1}\mu_v + 1 : \sum_{v=1}^{R}\mu_v - 1\big]$. From this and M_R, $\hat{X}_\cap(\mu_R)$ is retrieved. Let $\mathcal{J}^*$ denote the support of $\hat{X}_\cap(\mu_R)$, then E_1 evaluates $\mathcal{P}^* = \mathcal{P}_{alph}(\mathcal{J}^*)$.

Remark 5 *If $\mu_R = 1$ or $M_R = \mu_R$, then $\mathcal{P}^* = \mathcal{P}_{alph}(\mathcal{I}_R)$ is found directly without executing FindPSI-ring(M_R).*

Remark 6 *If $R = T$ and $\mu_T > 1$, then E_1 executes FindPSI-ring, without sending the two signaling symbols to the databases of E_N, because of the assumption of non-empty intersection, $\mathcal{P}$.*

Proof of Privacy: The privacy of the scheme follows from the individual privacy guarantees of the CarPSI-ring and FindPSI-ring primitives, performed on disjoint subsets of $\mathcal{P}_{alph}$. Moreover, each database receives queries of length μ_r for the rounds it is assigned, till R is found. This is regardless of the fact whether the value f_r is realized by any element in $\mathcal{P}_1$. Therefore, no information about the leader's function profile $\boldsymbol{\alpha}$ is leaked to any database. There is no leakage to the leader E_1, beyond $\mathcal{I}_{nom,1}$. After round $r < R$, E_1 only learns that

$\hat{X}_{\cap}(\mu_r)$ is empty and gains no information about the individual incidence vectors $\hat{X}_2([\mu_{R-1}])$ through $\hat{X}_N([\mu_{R-1}])$.

Communication Cost: The communication cost is contributed by R rounds of CarPSI-ring, and FindPSI-ring at the Rth round. First, we consider the case that $R < T$. Then, if $\mu_R > 1$,

$$C_{ring} = \sum_{r=1}^{R} (2(N-1)\mu_r + 2) + 2 + 2(\mu_R - 1) \tag{85}$$

$$= 2N \left(\sum_{r=1}^{R} \mu_r \right) + 2 \left(R - \sum_{r=1}^{R-1} \mu_r \right). \tag{86}$$

If $\mu_R = 1$, the cost reduces to $C_{ring} = 2N \left(\sum_{r=1}^{R} \mu_r \right) + 2 \left(R - \sum_{r=1}^{R} \mu_r \right)$ by Remark 5. Now, consider the case that $R = T$. If $\mu_R > 1$, following Remark 6, two fewer symbols are required, which gives

$$C_{ring} = 2N \left(\sum_{r=1}^{R} \mu_r \right) + 2 \left(R - \sum_{r=1}^{R-1} \mu_r \right) - 2 \tag{87}$$

$$= 2NK + 2 \left(R - 1 - \sum_{r=1}^{R-1} \mu_r \right), \tag{88}$$

since $\sum_{r=1}^{T} \mu_r = K$. If $\mu_R = 1$, no communication takes place, further saving $2\mu_R$ symbols and $C_{ring} = 2NK + 2 \left(R - 1 - \sum_{r=1}^{R} \mu_r \right)$. This completes the proof of Theorem 2.

5.3 Comparison with the Naive Scheme

First, we describe the PSI scheme for the ring setup, obtained by modifying the CarPSI-ring primitive.

PSI Scheme: In this scheme, the query generation process for entities E_1 through E_{N-1} matches that of CarPSI-ring(X_1) in Section 5.1.1, and differs in the answers returned by E_N. The databases of E_N generate answers from $\mathbb{F}_q^K$ as,

$$A_j = Q_{N-1,j} \circ X_N + S_N, \quad j = 1, 2. \tag{89}$$

The communication cost thus incurred is

$$C_{PSI,ring} = 2NK \tag{90}$$

symbols. The communication cost C_{ring} is upper bounded by $C_{PSI,ring}$, with P_{eq} denoting the probability that they are equal.

To evaluate P_{eq}, we study the conditions under which the communication costs are equal. It suffices to consider that $\mu_R > 1$, since C_{ring} is strictly lower if $\mu_R = 1$. First, if $R < T$ then, C_{ring} equals $2NK$ if and only if

$$2NK = 2N\sum_{r=1}^{R}\mu_r + 2\left(R - \sum_{r=1}^{R-1}\mu_r\right), \tag{91}$$

which implies

$$K - \sum_{r=1}^{R}\mu_r = \frac{R - \sum_{r=1}^{R-1}\mu_r}{N}, \tag{92}$$

which gives

$$\sum_{r=R+1}^{T}\mu_r = \frac{R - \sum_{r=1}^{R-1}\mu_r}{N}. \tag{93}$$

However, note that, in (93), the left-hand side is a positive integer, with value at least $T - R > 1$ since $\mu_r \geq 1$, $r \in [T]$, whereas, the right hand side is less than or equal to $\frac{1}{N}$, since the numerator is at most 1. This leads to a contradiction since $N \geq 2$. Thus, equality of costs happen only if $R = T$.

For $R = T$, if $\mu_R = \mu_T > 1$,

$$\begin{aligned} C_{ring} &= 2NK - 2\left(K - \mu_T - (T-1)\right) && (94)\\ &= 2NK + 2\left((T-1) - \sum_{r=1}^{T-1}\mu_r\right) && (95)\\ &\leq 2NK, && (96) \end{aligned}$$

since $\mu_r \geq 1$ by definition. Clearly, equality holds iff $\mu_r = 1, \forall r \in [T-1]$ and $\mu_T > 1$. Given the feasible sets $\mathcal{P}_1, \ldots, \mathcal{P}_N$ and τ, depending on the realization of f, T varies from 1 to $T_{\max}$ given by $T_{\max} = \min(\tau, K - \max(2, M) + 1)$. Since T is upper-bounded by τ, its maximum value when $\tau \leq K - \max(2, M) + 1$ is τ. Otherwise, the number of required communication rounds is $K - M + 1$, since $T = R$ if $M > 1$. If $M = 1$, the last communication round is skipped and $K - 1$ rounds suffice to find $\mathcal{P}^*$.

Next, we evaluate P_{eq} as follows:

$$P_{eq} = \mathbb{P}(R = T, \boldsymbol{\mu} = (1, 1, 1 \ldots, 1, \mu_T), \mu_T = K - (T-1) > 1) \tag{97}$$

$$= \sum_{r=1}^{T_{\max}} \mathbb{P}(T = r, \mu_1 = \mu_2 = \ldots = \mu_{r-1} = 1, \mu_r = K - (r-1)) \tag{98}$$

$$= \left(\sum_{r=1}^{T_{\max}} \sum_{j=1}^{\tau-1} \binom{\tau - j}{r-1} \right) \times \left(\frac{1}{\tau} \right)^K. \tag{99}$$

The values of P_{eq} for different values of τ and M are given in Table 2. The probability of equal communication costs is negligible and decreases as τ increases.

$P_{eq}(\times 10^{-3})$	$\tau = 2$	$\tau = 4$	$\tau = 6$	$\tau = 8$	$\tau = 10$
$M = 1$	2.93	1.43×10^{-2}	1.04×10^{-3}	2.37×10^{-4}	1.02×10^{-4}
$M = 2$	2.93	1.43×10^{-2}	1.04×10^{-3}	2.37×10^{-4}	1.02×10^{-4}
$M = 3$	2.93	1.43×10^{-2}	1.04×10^{-3}	2.37×10^{-4}	1.01×10^{-4}
$M = 4$	2.93	1.43×10^{-2}	1.04×10^{-3}	2.37×10^{-4}	9.67×10^{-5}

Table 2: P_{eq} with $K = 10$ for different values of τ and M.

6 The Case of $N > 2$ Entities: Star Setup

For the star setup, our DOFSP scheme is based on sequentially applying the PSI scheme of [11] on disjoint subsets of $\mathcal{P}_1$, determined by equi-cost elements. The leader E_1 chooses the retrieval indices from X_1 as $\mathcal{J}_1$ through $\mathcal{J}_R$ where f_{i_R} is the optimum function value in rounds 1 through $R \in [L]$. The operating field $\mathbb{F}_q$ should satisfy $q > N - 1$, where q is prime. To minimize the communication cost, q is chosen to be the minimum such prime that satisfies the above inequality.

6.1 Scheme Description

We illustrate this scheme with the same optimization setting as in Example 2.

Example 3 *Let $N_i = 2$ for all entities, then the candidates for l^* from (10) are $\{1, 3, 4\}$. Out of E_1, E_3 and E_4, we assign E_1 as the leader. Since $f(A) = 1$, $f(C) = f(D) = 5$, $f(G) = 3$ and $f(J) = 4$, we have $\mathcal{J}_1 = \{3, 4\}$, $\mathcal{J}_2 = \{10\}$, $\mathcal{J}_3 = \{7\}$ and $\mathcal{J}_4 = \{1\}$ with $\mathbb{F}_q$ such that $q = 5$. Then, the queries sent in the first round correspond to $\mathcal{J}_1$ and are*

$$Q_{i,1}^{(1)} = \{h_1, h_2\}, \quad Q_{i,2}^{(1)} = \{h_1 + e_3, h_2 + e_4\}, \quad i \in \{2, 3, 4\}, \tag{100}$$

where each $h_k \in \mathbb{F}_q^{10}$ $k \in \{1, 2\}$. The corresponding answers are

$$A_{i,1}^{(1)} = \{c(h_1^T X_i + S_{i,1}),\ c(h_2^T X_i + S_{i,2})\}, \tag{101}$$

$$A_{i,2}^{(1)} = \{c((h_1 + e_3)^T X_i + S_{i,1} + t_{i,2}(1)),\ c((h_2 + e_4)^T X_i + S_{i,2} + t_{i,2}(2))\}, \tag{102}$$

where $t_{2,2}(1)+t_{3,2}(1)+t_{4,2}(1)=t_{2,2}(2)+t_{3,2}(2)+t_{4,2}(2)=2$*, where* $S_{i,j}$*,* $j\in[2]$ *chosen uniformly at random from* $\mathbb{F}_5$ *by the respective* E_i *and* c *is picked uniformly at random from* $\mathbb{F}_5\setminus\{0\}$*. Then,* E_1 *subtracts the answers of the first database from the corresponding answers of the second database to get*

$$\{c(X_i(3)+t_{i,1}(k)),\ c(X_i(4)+t_{i,2}(k))\},\quad i\in\{2,3,4\},\ k\in\{1,2\}. \tag{103}$$

Adding these across i*,* E_1 *obtains*

$$Z_3 = c(X_2(3)+X_3(3)+X_4(3)+2)=3c, \tag{104}$$
$$Z_4 = c(X_2(4)+X_3(4)+X_4(4)+2)=2c, \tag{105}$$

where both $2c,3c\in\mathbb{F}_5\setminus\{0\}$*, implying that both* C *and* D *are absent in* $\mathcal{P}$*. The next best value is assumed by* J*, i.e.,* $\mathcal{J}_2=\{10\}$*. This time,* E_1 *sends the queries*

$$Q_{i,1}^{(2)}=h_3,\quad Q_{i,2}^{(2)}=h_3+e_{10},\quad i\in\{2,3,4\}, \tag{106}$$

and receives the answers

$$A_{i,1}^{(2)}=c(h_3^TX_i+S_{i,3}),\quad A_{i,2}^{(2)}=c((h_3+e_{10})^TX_i+S_{i,3}+t_{i,2}(3)). \tag{107}$$

By computing $A_{i,2}^{(2)}-A_{i,1}^{(2)}=c(X_i(10)+t_{i,2}(3))$ *for each* i *and adding them yields*

$$Z_{10}=c(X_2(10)+X_3(10)+X_4(10)+2)=3c, \tag{108}$$

which is an element of $\mathbb{F}_5\setminus\{0\}$*, implying that* J *does not belong to* $\mathcal{P}$*. The third best value is assumed by* G*, i.e.,* $\mathcal{J}_3=\{7\}$*. Next* E_1 *sends queries*

$$Q_{i,1}^{(3)}=h_4,\quad Q_{i,2}^{(3)}=h_4+e_7, \tag{109}$$

and receives answers

$$A_{i,1}^{(3)}=c(h_4^TX_i+S_{i,4}),\quad A_{i,2}^{(3)}=c((h_4+e_7)^TX_i+S_{i,4}+t_{i,2}(4)). \tag{110}$$

for $i\in\{2,3,4\}$ *and processes them as before to obtain*

$$Z_7=c(X_2(7)+X_3(7)+X_4(7)+2)=c(3+2)=5c, \tag{111}$$

which is 0 *in* $\mathbb{F}_5$*. This implies that* G *is an element of* $\mathcal{P}$*, hence,* $\mathcal{P}^*=\{G\}$*. There are a total of* 8 *queries sent to each of* E_2*,* E_3*,* E_4*, each of which is a* 10*-length vector of* $\mathbb{F}_q$*, resulting in* $U_{star}=3\cdot8\cdot10=240$*, and there are* 8 *corresponding answers, giving* $D_{star}=2(N-1)(\alpha_1+\alpha_2+\alpha_3)=24$*, hence* $C_{star}=(10+1)D_{star}=264$*.*

Now, if $N_i = 3$ for the non-leader entities, the queries sent $i \in \{2,3,4\}$ are

$$Q_{i,1}^{(1)} = h_1, Q_{i,2}^{(1)} = h_1 + e_3, Q_{i,3}^{(1)} = h_1 + e_4, \tag{112}$$
$$Q_{i,1}^{(2)} = h_2, Q_{i,2}^{(2)} = h_2 + e_{10}, \tag{113}$$
$$Q_{i,3}^{(3)} = h_2 + e_7, \tag{114}$$

and the corresponding answers are as follows,

$$A_{i,1}^{(1)} = c(h_1^T X_i + S_{i,1}), \tag{115}$$
$$A_{i,2}^{(1)} = c((h_1 + e_3)^T X_i + S_{i,1} + t_{i,2}(1)), \tag{116}$$
$$A_{i,3}^{(1)} = c((h_1 + e_4)^T X_i + S_{i,1} + t_{i,3}(1)), \tag{117}$$
$$A_{i,1}^{(2)} = c(h_2^T X_i + S_{i,2}), \tag{118}$$
$$A_{i,2}^{(2)} = c((h_2 + e_{10})^T X_i + S_{i,2} + t_{i,2}(2)), \tag{119}$$
$$A_{i,3}^{(3)} = c((h_2 + e_7)^T X_i + S_{i,2} + t_{i,3}(2)), \tag{120}$$

giving $D_{star} = 3\lceil \frac{3\times 3}{3-1} \rceil = 15$ and $C_{star} = (10+1) \cdot D_{star} = 165$, which is lower than the $N_i = 2$ case.

We now describe the steps of the general achievable scheme.

Assignment of Common Randomness: The randomness symbols are assigned for P_1 indices to each of the non-leader databases, prior to the start of communication in the same manner as in [11].

- First, $\lceil \frac{P_1}{N_i - 1} \rceil$ *local common randomness* symbols uniformly picked from $\mathbb{F}_q$ are shared locally among the databases of the respective non-leader entities. E_i shares the set $\mathcal{C}_{loc,i} = \{S_{i,1}, \ldots, S_{i,m_i}\}$ where $m_i = \lceil \frac{P_1}{N_i - 1} \rceil$ $i \in [2:N]$.

- Then, each database generates a set of *correlated randomness* symbols from $\mathbb{F}_q$. Specifically, database $j \in [2:N_i]$ of E_i generates the set $\mathcal{C}_{cor,i,j} = \{t_{i,j}(1), t_{i,j}(2), \ldots, t_{i,j}(l_{i,j})\}$, where $l_{i,j} = \lceil \frac{P_1 - j + 1}{N_i - 1} \rceil$ with $t_{i,1}(k) = 0$, such that, every $j \in [2:N_i]$, letting

$$\hat{t}_{i,l} = t_{i,j}(k), \quad l = (k-1)(N_i - 1) + (j-1), k \in \left\{1, \ldots, \left\lceil \frac{P_1 - j + 1}{N_i - 1} \right\rceil \right\}, \tag{121}$$

  the following equality

$$\sum_{i=2}^{N} \hat{t}_{i,l} = q - (N-1) \tag{122}$$

  is satisfied, for all $l \in [P_1]$.

- Finally, the *global common randomness* symbol c chosen uniformly at random from $\mathbb{F}_q \setminus \{0\}$ is known to all entities except E_1.

Thus, for each $i \in [2:N]$, $\mathcal{S}_i$ is comprises $\mathcal{C}_{loc,i}, \mathcal{C}_{corr,i,j}, j \in [2:N_i]$ and c.

Query and Answer Generation: The query and answer generation steps for each round r corresponds to those of the PSI scheme in [11] with the subsets $\alpha_r, r \in [L]$ till $r = R$. We describe it here briefly as follows. We write, $\mathcal{J}_r = \{u_{r,1}, u_{r,2}, \ldots, u_{r,\alpha_r}\}$ with $u_{r,1} < u_{r,2} < \ldots < u_{r,\alpha_r}$.

In the first round, E_1 generates $k_1 = \max_{i\in[2:N]} \lceil \frac{\alpha_1}{N_i - 1} \rceil$ vectors $h_1, h_2, \ldots, h_{k_1}$ independently and uniformly at random from $\mathbb{F}_q^K$. The queries sent to database 1 of E_i are

$$Q_{i,1}^{(1)} = \left\{h_1, h_2, \ldots, h_{v_{1,i}}\right\}, \quad v_{1,i} = \left\lceil \frac{\alpha_1}{N_i - 1} \right\rceil, \quad i \in [2:N]. \tag{123}$$

Now, $\alpha_1 = (v_{1,i}-1)(N_i-1)+l_{i,1}$ with $l_{i,1} < N_i - 1$. The queries sent to databases $j \in [2:N_i]$ of E_i for $i \in [2:N]$ are

$$Q_{i,j}^{(1)} = \begin{cases} \{h_k + e_{u_{1,m}}, k \in [v_{1,i}], m = (k-1)(N-1) + j - 1\}, & j \in [2 : l_{i,1} + 1], \\ \{h_k + e_{u_{1,m}}, k \in [v_{1,i} - 1], m = (k-1)(N-1) + j - 1\}, & j \in [l_{i,1} + 1 : N_i]. \end{cases} \tag{124}$$

The answers that the databases of $E_i, i \in [2:N]$ respond with are,

$$A_{i,j}^{(1)} = \begin{cases} \{c(h_k^T X_i + S_{i,k}), k \in [v_{1,i}]\}, & j = 1, \\ \{c((h_k + e_{u_{1,m}})^T X_i + S_{i,k} + t_{i,j}(k)), m = (k-1)(N-1) + j - 1\}, & j \in [2:N_i], \end{cases} \tag{125}$$

where the values of k, for $j \in [2:N_i]$ are the same as those in (124). Using these answers, the leader decodes $X_\cap(u_{1,m})$ for each $m \in [\alpha_1]$. Let $t_{1,j}(k)$ map to $\hat{t}_{i,l}, l \in [P_1]$ given by (121), then E_1 computes, for each answer from E_i, the set of differences, $\forall j \in [2:N_i]$,

$$A_{1,j}^{(1)} - A_{1,1}^{(1)} = \{c((h_k + e_{u_{1,m}})^T X_i + S_{i,k} + \hat{t}_{i,l} - c(h_k^T X_i + S_{i,k})\} \tag{126}$$

$$= \{c(X_i(u_{1,m}) + \hat{t}_{i,l})\}. \tag{127}$$

Next, using the answers from all $E_{2:N}$, for each $u_{1,m}$, E_1 computes the following sum in $\mathbb{F}_q$,

$$Z_{u_{1,m}} = \sum_{i=2}^{N} c(X_i(u_{1,m}) + \hat{t}_{i,l}) \tag{128}$$

$$= c\left(\sum_{i=2}^{N} X_i(u_{1,m})\right) + q - (N-1), \tag{129}$$

from (122). Now, if $\sum_{i=2}^{N} X_i(u_{1,m}) = N-1$, $Z_{u_{1,m}} = 0$, which implies $X_i(u_{1,m}) = 1$ for each $i \in [2:N]$ and $\mathcal{P}_{alph}(u_{1,m}) \in \mathcal{P}$. Otherwise, if $Z_{u_{1,m}} \in \mathbb{F}_q \setminus \{0\}$, then $X_i(u_{1,m}) = 0$ for some $i \in [2:N]$ and $\mathcal{P}_{alph}(u_{1,m}) \notin \mathcal{P}$. If $Z_{u_{1,m}} \neq 0$ for every $m \in [\alpha_1]$, then $X_\cap$ has a 0 at all indices in $\mathcal{J}_1$, and the scheme proceeds to round 2.

In round $r \geq 2$, the queries and answers are denoted by $Q_{i,j}^{(r)}$ and $A_{i,j}^{(r)}$, respectively. These are used to retrieve $X_\cap(\mathcal{J}_{r,m}) = \{X_\cap(u_{r,1}), \ldots, X_\cap(u_{r,\alpha_r})\}$. Similar to the DOFSP scheme for $N = 2$, queries are assigned using a new random vector or by reusing the previous one, depending on whether the last query was sent to database N_i or database $j \in [2 : N_i - 1]$, respectively. The same procedure as $r = 1$ is followed for both query and answer generation.

Proof of Privacy: The privacy guarantee of DOFSP is a direct consequence of the privacy of PSI scheme given in [11]. In each round, every database receives as query, a K-dimensional vector chosen uniformly from $\mathbb{F}_q^K$. Each database performs the same computation to determine its answer.

Communication Cost: Clearly, round r is equivalent to conducting the PSI on $\mathcal{P}_{alph}(\mathcal{J}_r)$ and the scheme proceeds till $r = R$ where $Z_{u_{R,m}} = 0$ for some $l \in \alpha_R$ for the first time. Let $\mathcal{J}^* = \{u_{R,m} : Z_{u_{R,m}} = 0\}$, then the solution set is $\mathcal{P}^* = \mathcal{P}_{alph}(\mathcal{J}^*)$. The scheme is an iterative implementation of the PSI on $\sum_{r=1}^{R} \alpha_r$ elements, and hence, results in the download cost,

$$D_{star} = \sum_{i=2}^{N} \left\lceil \frac{(\sum_{r=1}^{R} \alpha_r) N_i}{N_i - 1} \right\rceil. \tag{130}$$

The total communication cost is, therefore, $C_{star} = (K+1)D_{star}$ symbols of $\mathbb{F}_q$. The entities cannot share prior knowledge of $\boldsymbol{\alpha}$ as it would violate E_1's privacy constraint. Hence, the communication cost depends heavily on the choice of E_1 and is upper-bounded by $\sum_{i=2}^{N} \lceil \frac{P_1 N_i}{N_i - 1} \rceil$.

Remark 7 *Similar to the $N = 2$ case, E_1 skips the PSI step for $\mathcal{J}_R$ if $R = L$ and $\alpha_L = 1$, in which case $D_{star} = \sum_{i=2}^{N} \left\lceil \frac{(\sum_{r=1}^{L-1} \alpha_r) N_i}{N_i - 1} \right\rceil$.*

6.2 Comparison with the Naive Scheme

Note that the condition under which $\mathcal{I}_{nom,i}$ for entities $E_i, i \in [2:N]$ is equal to that of PSI is when all the elements of $\mathcal{P}_1$ have been checked for their presence in $\mathcal{P}$, hence E_1 learns $\mathcal{P}^* = \mathcal{P}$. This occurs whenever $R = L$, i.e., when the optimal value of f is f_{i_L}, realized by the M elements in $\mathcal{P}^*$. In terms of communication cost, it suffices to compare the download cost D_{star} with $D_{PSI,star}$ in [11], which is,

$$D_{PSI,star} = \sum_{i=2}^{N} \left\lceil \frac{P_1 N_i}{N_i - 1} \right\rceil, \tag{131}$$

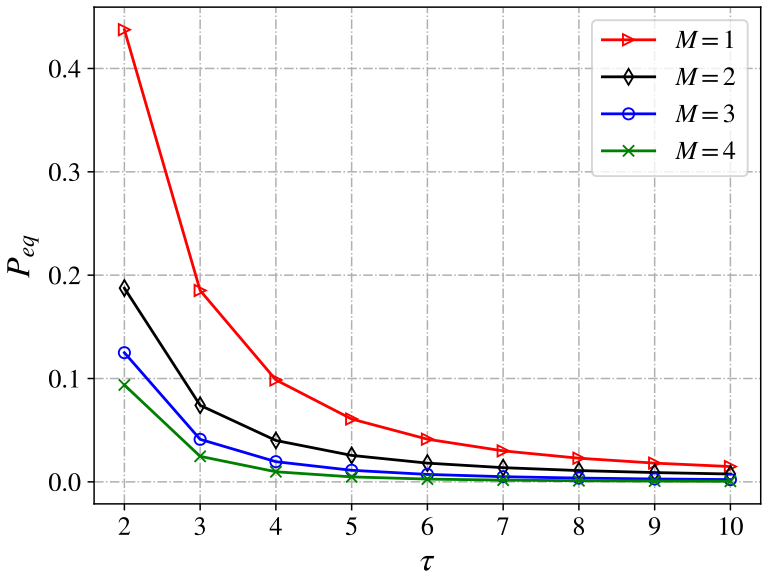


Figure 4: P_{eq} against τ for various values of M with $N > 2$.

under the setting where E_1 is the leader. We have $D_{star} < D_{PSI,star}$ and equal only when $\sum_{r=1}^{R} \alpha_r = P_1$, i.e., when the search for $\mathcal{P}^*$ continues till round $R = L$ and $\alpha_L > 1$. Note that L depends on the realization of f and given the feasible sets $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_N$, its maximum value is $L_{\max} = \min(\tau, P_1 - \max(2, M) + 1)$. For $\tau \leq P_1 - \max(2, M) + 1$, indeed the maximum value of L is τ. For $\tau > P_1 - \max(2, M) + 1$, this can be verified as follows. If $M = 1$, the maximum number of rounds that the communication takes place is $P_1 - 1$, when $\alpha_r = 1$, $r \in [L]$ throughout which the set intersection is empty. Otherwise, the maximum number of rounds is $L = P_1 - M + 1$, when $\alpha_r = 1$, $r \in [L]$ throughout which the set intersection is empty.

Under uniform mapping of f, P_{eq} is computed as follows,

$$P_{eq} = \mathbb{P}\left(\sum_{r=1}^{L_{\max}} \alpha_r = P_1, \alpha_L > 1\right) \tag{132}$$

$$= \left(\frac{1}{\tau}\right)^{P_1} \times \sum_{r=1}^{L_{\max}} \mathcal{N}_{\boldsymbol{\alpha}}(r), \tag{133}$$

where

$$\mathcal{N}_{\boldsymbol{\alpha}}(r) = \sum_{j=1}^{\tau-1} \binom{\tau - j}{r - 1} \sum_{\alpha_1 \geq 1, \ldots, \alpha_{r-1} \geq 1, \alpha_r \geq \max(M,2): \alpha_1 + \alpha_2 + \ldots + \alpha_r = P_1} \binom{P_1 - M - r}{\alpha_1, \alpha_2, \ldots, \alpha_r - M}. \tag{134}$$

Given a fixed P_1 for the leader set, under varying sizes of τ and intersection sizes M, we plot the probability P_{eq} in Fig. 4. Note the increase in P_{eq} compared to the $N = 2$ case given any τ and M. This is because, for the two-user case, D_{star} is always an upper-bound on D.

7 Conclusion

In this work, we proposed an information-theoretic formulation of constrained distributed optimization problem with multiple agents subject to a privacy constraint, i.e., distributed optimization with feasible set privacy (DOFSP). Our DOFSP formulation is closely connected to several private multi-set operations, such as, private set intersection (PSI), PSI cardinality (CarPSI) and threshold PSI (ThPSI), which are of independent interest. The main idea of our formulation is to assign a single leader agent to coordinate the communications, with non-colluding databases of the non-leader agents. At the same time, these databases, with shared randomness, allow zero leakage of information on the feasible sets that they store. As our main result, we show that, compared to a naive scheme where the entities invoke a PSI protocol to learn the entire set intersection, and then compute the solution, our DOFSP scheme is more efficient. Our scheme saves the communication cost with a high probability, and most importantly, prevents additional leakage about the set intersection beyond the solution set.

For $N = 2$, and the star setup, we adopt an SPIR framework for our DOFSP schemes. Unlike the $N = 2$ case which evaluates the set cardinalities before finding the intersection, our scheme sequentially finds set intersections for the star setup. This increases the communication cost for $N > 2$ since the leader does not know apriori whether the intersection is empty. While the assignment of the leader minimizes the worst-case communication cost for the DOFSP problem, it is not optimized specifically for the latter. On the contrary, the communication cost in the ring setup is independent of the choice of leader. Thus, developing a more efficient DOFSP scheme for the star setup, is open for further investigation.

For $N > 2$ under the ring setup, we utilize secret-sharing and one-time padding to develop our primitives. We observe that the PSI scheme here operates differently from that in [11] adopted in the star setup. The communication cost in the ring setup scales linearly with N and K and is independent of P_1, the leader set size, while that in the star setup grows linearly with P_1, along with N and K. However, the communication cost of the scheme in [11] reduces with more databases per entity, while that in the ring setup is unaffected by this quantity.

In our formulation, we did not make any assumption on the objective function f. However, restricting to specific classes of functions can induce special structure on $\boldsymbol{\mu}$. This can possibly improve the communication cost by the design of schemes tailored to this structure. Furthermore, it would be interesting to consider more general communication setups, apart from star and ring, considered in this paper. Finally, extension to provide resilience against other threat models, through information-theoretic privacy is another promising direction. In our work, we assumed honest agents, who do not collude among themselves. It would be interesting to study the DOFSP problem with colluding agents that share information and try to breach the privacy of the remaining agents.

References

[1] S. Han, U. Topcu, and G. J. Pappas. Differentially private distributed constrained optimization. *IEEE Transactions on Automatic Control*, 62(1):50–64, January 2017.

[2] A. Nedic, A. Ozdaglar, and P. A. Parrilo. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4):922–938, April 2010.

[3] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, August 2014.

[4] Z. Huang, S. Mitra, and N. Vaidya. Differentially private distributed optimization. In *ICDCN*, January 2015.

[5] M. Showkatbakhsh, C. Karakus, and S. Diggavi. Differentially private consensus-based distributed optimization. Available online at arXiv:1903.07792.

[6] N. Gupta, S. Gade, N. Chopra, and N. H. Vaidya. Preserving statistical privacy in distributed optimization. *IEEE Control Systems Letters*, 5(3):779–784, June 2021.

[7] T. Ding, J. He, and X. Guan. Distributed optimization via state and direction perturbation in multiagent systems. *IEEE Transactions on Automatic Control*, 67(2):722–737, Feb 2022.

[8] J. Hsu, A. Roth, T. Roughgarden, and J. Ullman. Privately solving linear programs. In *ICALP*, July 2014.

[9] Y. Lu and M. Zhu. Privacy preserving distributed optimization using homomorphic encryption. *Automatica*, 96:314–325, 2018.

[10] Z. Wang, K. Banawan, and S. Ulukus. Private set intersection: A multi-message symmetric private information retrieval perspective. *IEEE Transactions on Information Theory*, 68(3):2001–2019, March 2022.

[11] Z. Wang, K. Banawan, and S. Ulukus. Multi-party private set intersection: An information-theoretic approach. *IEEE Journal on Selected Areas in Information Theory*, 2(1):366–379, March 2021.

[12] H. Sun and S. A. Jafar. The capacity of private information retrieval. *IEEE Transactions on Information Theory*, 63(7):4075–4088, July 2017.

[13] K. Banawan and S. Ulukus. The capacity of private information retrieval from coded databases. *IEEE Transactions on Information Theory*, 64(3):1945–1956, January 2018.

[14] C. Tian, H. Sun, and J. Chen. Capacity-achieving private information retrieval codes with optimal message size and upload cost. *IEEE Transactions on Information Theory*, 65(11):7613–7627, November 2019.

[15] S. Ulukus, S. Avestimehr, M. Gastpar, S. A. Jafar, R. Tandon, and C. Tian. Private retrieval, computing, and learning: Recent progress and future challenges. *IEEE Journal on Selected Areas in Communications*, 40(3):729–748, March 2022.

[16] S. Vithana, Z. Wang, and S. Ulukus. Private information retrieval and its extensions: An introduction, open problems, future directions. *IEEE BITS the Information Theory Magazine*, 3(4):67–85, December 2023.

[17] H. Sun and S. A. Jafar. The capacity of symmetric private information retrieval. *IEEE Transactions on Information Theory*, 65(1):322–329, January 2018.

[18] E. D. Cristofaro and G. Tsudik. Practical private set intersection protocols with linear complexity. In *Proc. Int. Conf. on Fin. Crypt. and Data Sec.*, page 143–159. Springer-Verlag, January 2010.

[19] E. Zhang, J. Chang, and Y. Li. Efficient threshold private set intersection. *IEEE Access*, 9:6560–6570, 2021.

[20] S. Ghosh and M. Simkin. The communication complexity of threshold private set intersection. In *CRYPTO 2019*, pages 3–29. Springer, Cham, August 2019.

[21] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, November 1979.

[22] C. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28(4):656–715, 1949.