Private Set Intersection Using Multi-Message Symmetric Private Information Retrieval

Zhusheng Wang\(^1\), Karim Banawan\(^2\), and Sennur Ulukus\(^1\)

\(^1\)Department of Electrical and Computer Engineering, University of Maryland, College Park, MD, USA
\(^2\)Electrical Engineering Department, Faculty of Engineering, Alexandria University, Alexandria, Egypt

Abstract—We study the problem of private set intersection (PSI). In PSI, there are two entities, each storing a set \(P_i\), whose elements are picked from a finite set \(S_K\), on \(N_i\) replicated and non-colluding databases. It is required to determine the set intersection \(P_1 \cap P_2\) without leaking any information about the remaining elements to the other entity. We first show that the PSI problem can be recast as a multi-message symmetric private information retrieval (MM-PIR) problem. Next, as a stand-alone result, we show that the exact capacity of MM-PIR is \(C_{MM-PIR} = 1 - \frac{1}{N}\) when \(P_i \leq K - 1\), if the common randomness \(S\) satisfies \(H(S) \geq \frac{N - 1}{N}P\) per desired symbol. This result implies that there is no gain for MM-PIR over successive single-message SPIR. We present a novel capacity-achieving scheme which builds seamlessly over the multi-message PIR (MM-PIR) scheme. Based on this capacity result for the MM-PIR problem, we show that the optimal download cost for the PSI problem is given by \(\min \left\{ \frac{P_1 N_2}{N_2 - 1}, \frac{P_2 N_1}{N_1 - 1} \right\}\), where \(P_i\) is the cardinality of the set \(P_i\).

I. INTRODUCTION

Private set intersection (PSI) refers to the problem of determining the common elements in two sets without leaking any further information about the remaining elements in the sets. PSI has been a major research topic in the field of cryptography starting with [1]. As a practical motivation for PSI, consider an airline company which has a list of its customers and a law enforcement agency revealing the respective lists without the airline company revealing the rest of the suspects in its list (see also [2], [3]).

Since the entities in PSI want to privately retrieve the elements that belong to the intersection, private information retrieval (PIR) can be a building block for the PSI problem [4]. Nevertheless, it is needed to keep the remaining elements of the sets secret from the other entity. This gives rise naturally to the problem of symmetric PIR (SPIR), which was originally introduced in [5]. Recently, Sun and Jafar reformulated the problems of PIR and SPIR from an information-theoretic point of view, and determined the fundamental limits of both of these problems, in [6] and [7], respectively. Subsequently, the fundamental limits of many interesting variants of PIR and SPIR have been considered, see for example [8]–[51].

To use SPIR to implement PSI, the \(i\)th entity needs to privately check the presence of each element in its set at the other entity. Hence, the \(i\)th entity needs to retrieve multiple messages from the other entity, where the messages correspond to the incidences of each element of the set. This establishes the connection between PSI and multi-message SPIR (MM-SPIR). The MM-SPIR problem is interesting on its own right and has remained an open problem until this work. The papers that are most closely related to our work are the ones that focus on symmetry and multi-message aspects of PIR as in [7], [11]–[15], [20], [29]–[31]. None of these works considers the interplay between the data privacy constraint and the joint retrieval of multiple messages, as needed in MM-SPIR.

In this paper, first focusing on MM-SPIR as a stand-alone problem, we derive its capacity to be \(C_{SM-PIR} = C_{MM-PIR} = 1 - \frac{1}{N}\). We show that the databases need to share a random variable \(S\) such that \(H(S) \geq \frac{N - 1}{N}P\) per desired symbol. This implies that, unlike MM-PIR, there is no gain from jointly retrieving the \(P\) messages. For the extreme case \(P = K\), full capacity is attained without the need for any common randomness. Further, we propose a novel capacity-achieving scheme for \(1 \leq P \leq K - 1\). The query structure of the scheme resembles its counterpart in [20]. Our scheme is surprisingly optimal for all \(P\) and \(K\) in contrast to the scheme in [20] which is proved to be optimal only if \(P\) is at least half of \(K\) or \(K/P\) is an integer. By plugging \(P = 1\), our scheme serves as an alternative capacity-achieving scheme for the SM-SPIR scheme in [7]. As an added advantage, our scheme extends seamlessly the MM-PIR scheme to satisfy the database privacy constraint without changing the query structure. Hence, by operating such a scheme the databases can support SPIR and PIR simultaneously.

We ultimately consider the PSI problem. There are two entities \(E_1\) and \(E_2\). The entity \(E_i\) has a set (list) \(P_i\), whose elements are picked from a finite set \(S_K\) and has a cardinality \(P_i\). The set \(P_i\) is stored on \(N_i\) non-colluding and replicated databases. It is required to compute the intersection \(P_1 \cap P_2\) without leaking information about \(P_1 \setminus P_2\) or \(P_2 \setminus P_1\) with the minimum download cost. We first show that this problem can be recast as an MM-SPIR problem, where a user needs to retrieve \(P\) messages from a library containing \(K\) messages. In this MM-SPIR problem, messages correspond to incidences of elements in these sets with respect to the field elements. The incidence vector is a binary vector of length \(K\) that stores
a 1 in the position of the $j$th element of the field if this field element is in $P_i$. This transforms each set into a library of $K$ binary messages (of length 1 bit each). Therefore, in transforming the PSI problem into an MM-SPIR problem, two restrictions arise: First, the message size is 1 bit. Second, in our formulation, we restrict the set generation model to the case where the resulting messages are independent. Following these constructions, entity $E_1$ performs MM-SPIR of the messages corresponding to its set $P_1$ within the databases of the other entity. By decoding these messages, the intersection $P_1 \cap P_2$ is determined without leaking any information about $P_1 \setminus P_2$ or $P_2 \setminus P_1$. We show that the optimal download cost for the PSI problem is \[ \min \left\{ \left\{ \frac{P_1 N_2}{N_2 - 1}, \frac{P_2 N_1}{N_1 - 1} \right\} \right\} \]. The linear scaling of the download cost appears in the problem of determining the set intersection even without any privacy constraints. We only provide sketches of the proofs here due to space limitations; proof details, examples and figures can be found in [52].

II. PSI: Problem Formulation

Consider a setting with two entities $E_1$ and $E_2$ each storing a set $P_i$, $i = 1, 2$. For each element of the finite set $\mathbb{S}_K$, the entity $E_i$ adds this element to its set $P_i$ independently from the remaining field elements with probability $q_i$. We focus on the case of $q_i = \frac{1}{2}$ for $i = 1, 2$. After generation of the set $P_i$, the cardinality of $P_i \subseteq \mathbb{S}_K^P$ is denoted by $|P_i| = P_i$, and is public knowledge. The entity $E_i$ stores $P_i$ in a replicated fashion on $N_i$ non-colluding databases.

The entities $E_1$ and $E_2$ want to compute the intersection $P_1 \cap P_2$ privately. To that end, $E_1$ sends the query $Q_{n_2}^{P_1}$ to the $n_2$th database (which is associated to $E_2$) for all $n_2 \in [N_2]$, where $[N_2]$ (and also $[1 : N_2]$) denotes integers from 1 to $N_2$. Since $E_1$ does not know $P_2$ in advance, it generates the queries $Q_{1:N_2}^{P_1} = \{ Q_{n_2}^{P_1} : n_2 \in [N_2] \}$ independently from $P_2$, hence,

$$ I(Q_{1:N_2}^{P_1}; P_2) = 0 \quad (1) $$

The databases associated with $E_2$ respond with answer strings $A_{1:N_2}^{P_1}$. The $n_2$th answer string $A_{n_2}^{P_1}$ is a deterministic function of the set $P_2$ and the query $Q_{n_2}^{P_1}$, thus,

$$ H(A_{n_2}^{P_1} | Q_{n_2}^{P_1}, P_2) = 0, \quad n_2 \in [N_2] \quad (2) $$

$E_1$ should be able to reliably compute the intersection $P_1 \cap P_2$ based on the collected answer strings $A_{1:N_2}^{P_1}$, i.e.,

$$ H(P_1 \cap P_2 | A_{1:N_2}^{P_1}, A_{1:N_2}^{P_1}) = 0 \quad (3) $$

For privacy, first, the queries sent by $E_1$ should not leak any information about $P_1$, i.e., any individual database associated with $E_2$ learns nothing about $P_1$ from the queries.

$E_1$ should not learn anything further than $P_1 \cap P_2$ from $E_2$ based on the collected answer strings. Thus, $E_1$ should learn nothing about the information contained in the subset $(P_2 \setminus P_1) \cup (P_1 \cup P_2) = P_1$ of $E_2$.

$$ I(P_1; A_{1:N_2}^{P_1}) = 0 \quad (4) $$

Second, $E_1$ should not learn anything further than $P_1 \cap P_2$ from $E_2$ based on the collected answer strings. Thus, $E_1$ should learn nothing about the information contained in the subset $(P_2 \setminus P_1) \cup (P_1 \cup P_2) = P_1$ of $E_2$.

$$ I(P_1; A_{1:N_2}^{P_1}) = 0 \quad (5) $$

An achievable PSI scheme should satisfy the PSI reliability constraint (3), the $E_1$ privacy constraint (4), and the $E_2$ privacy constraint (5). The efficiency of the scheme is measured by the total number of downloaded bits by one of the entities $E_1$ or $E_2$ in order to compute $P_1 \cap P_2$, denoted by $D$,

$$ D = \sum_{n_2=1}^{N_2} H(A_{n_2}^{P_1}) \quad (6) $$

The optimal download cost is $D^* = \inf D$ over all achievable PSI schemes.

III. From PSI to MM-SPIR

We show that PSI can be reduced to an MM-SPIR problem, if the entities allow storing their sets in a specific searchable format. This transformation has the same flavor as [53] and [35]. This enables PIR, which assumes that the user knows the position of the desired file in the databases. Define the incidence vector $X_i \in \mathbb{F}_2^N$ as a binary vector of size $K$ associated with the set $P_i$. Denote the $j$th element of the incidence vector $X_i$ by $X_i(j)$ where $X_i(j) = 1$ if $j \in P_i$ for all $j \in \mathbb{S}_K$. Hence, $X_i(j)$ is an i.i.d. random variable for all $j \in \mathbb{S}_K$ such that $X_i(j) \sim \text{Ber}(q_i)$. The entity $E_i$ constructs the incidence vector $X_i$ corresponding to the set $P_i$ and replicates $X_i$ at all of its $N_i$ associated databases. The PSI determination process is performed over $X_1$ or $X_2$.

Assume that $E_1$ initiates the PSI process. $E_1$ does not know $M$, the size of the intersection, in advance. The only information $E_1$ has is $P_1$. Consequently, $E_1$ wants to verify the existence of each element of $P_1$ in $P_2$ to deduce $P_1 \cap P_2$. Thus, $E_1$ needs to jointly and reliably download the bits $W_{P_1} = \{ X_2(j) : j \in P_1 \}$ by sending $N_2$ queries to the databases associated with $E_2$ and collecting the corresponding answer strings. Hence, the reliability constraint is,

$$ H(W_{P_1}, Q_{1:N_2}^{P_1}, A_{1:N_2}^{P_1}) = 0 \quad (7) $$

Since $E_1$ is searching for the existence of all elements of $P_1$ in $P_2$ without leaking any information about $P_1$ to any individual database associated with $E_2$, the $E_1$ privacy constraint in (4) dictates,

$$ I(P_1; Q_{n_2}^{P_1}) = 0, \quad n_2 \in [N_2] \quad (8) $$

This is the privacy constraint in the MM-PIR problem [20].

To ensure the $E_2$ privacy constraint, the answers from $E_2$ databases should not leak any information contained in the subset $P_1$ of $E_2$, which is equivalent to not leaking any information about $W_{P_1}$,

$$ I(W_{P_1}; A_{1:N_2}^{P_1}) = 0 \quad (9) $$

This is exactly the database privacy constraint in MM-SPIR.

Consequently, the PSI problem reduces to MM-SPIR with i.i.d. messages of length 1 bit each, if the entities $E_1$ and $E_2$ are allowed to construct the corresponding incidence vectors for the original sets. In Section V, we derive in detail the capacity of the MM-SPIR problem, which in turn gives the most efficient information-theoretic PSI scheme.
IV. Main Result

Our main result provides the optimal download cost for the PSI problem under the assumptions in Sections II and III.

**Theorem 1** In the PSI problem, if the elements of the sets are added independently with probability $q_i = \frac{1}{N}$ from a finite set of size $K$, and if the set $P_1$ where $|P_1| = P_1$ is stored among $N_1$ databases and the set $P_2$ where $|P_2| = P_2$ is stored among $N_2$ databases, then the optimal download cost is,

$$D^* = \min \left\{ \left[ \frac{P_1 N_2}{N_2 - 1} \right], \left[ \frac{P_2 N_1}{N_1 - 1} \right] \right\} \quad (10)$$

The proof of Theorem 1 is a direct consequence of the capacity result for MM-SPIR presented in Section V. We note that compared with the best-known result for PSI under computational guarantees [2], assuming that $P_1 \geq P_2$, and $N_1 = N_2$, our result $O(P_2)$ is only linear in the size of the smaller set in contrast to $O(P_2 \log P_1)$ in [2]. The linear scalability of our scheme matches the linear scalability of the best-known set intersection algorithms without any privacy constraints. Our result is private in information-theoretic (absolute) sense and does not need any assumptions about the computational powers of the entities. Furthermore, the achievable scheme is fairly simple and easy to implement compared to the fully homomorphic encryption needed in [2]. The only drawback of our approach is that it needs multiple non-colluding databases ($N_1$ or $N_2$ needs to be strictly larger than 1), otherwise, our scheme is infeasible.

V. MM-SPIR as a Stand-Alone Problem

In this section, we consider the MM-SPIR problem. We present the problem in a stand-alone format, i.e., we present a formal problem description in Section V-A, followed by the main result in Section V-B, the converse in Section V-C, and a novel achievability in Section V-D.

A. MM-SPIR: Formal Problem Description

There are $N$ non-colluding databases storing $K$ i.i.d. messages. Each message is composed of $\ell$ i.i.d. and uniformly chosen symbols from a sufficiently large finite field, i.e., $H(W_k) = L$ for $k \in [K]$ and $H(W_{1,K}) = KL$.

In MM-SPIR, the user needs to retrieve $W_P$, with the desired message set $P$ having size $|P| = P$. Following [7], let $F$ denote the private randomness possessed by the user to satisfy the user privacy constraint. A necessary common randomness $S$ must be shared among the $N$ databases to satisfy the database privacy constraint. $S$ and $F$ are generated independently, and independent of the message set $W_{1,K}$ without knowing the desired index set $P$. Then,

$$H(F, S, P, W_{1,K}) = H(F) + H(S) + H(P) + H(W_{1,K}) \quad (11)$$

To perform MM-SPIR, a user generates a query $Q_{n}^{[P]}$ and sends it to the $n$th database. Hence, the queries $Q_{1:N}^{[P]}$ are deterministic functions of $F$, i.e.,

$$H(Q_{1:N}^{[P]}, F) = 0, \quad \forall P \quad (12)$$

From (11) and (12), the queries are independent of $W_{1,K}$, i.e.,

$$I(Q_{1:N}^{[P]}; W_{1,K}) = 0 \quad (13)$$

After receiving a query from the user, each database truthfully generates an answer string based on the messages and the common randomness, hence,

$$H(A_n^{[P]}|Q_{1:N}^{[P]}, W_{1,K}, S) = 0, \quad \forall n, \forall P \quad (14)$$

After collecting all the answer strings from the $N$ databases, the user should be able to decode $W_P$ reliably, therefore,

$$[\text{reliability}] \quad H(W_P|A_n^{[P]}, Q_{1:N}^{[P]}, F) = 0 \quad \forall P \quad (15)$$

The user privacy constraint can be written as,

$$[\text{user privacy}] \quad I(P; Q_{n}^{[P]}, A_n^{[P]}, W_{1,K}, S) = 0, \quad \forall n, P \quad (16)$$

In order to protect the databases’ privacy, the user should learn nothing about $W_P$ which is the complement of $W_P$,

$$[\text{database privacy}] \quad I(W_P; Q_n^{[P]}, A_n^{[P]}, F) = 0, \quad \forall P \quad (17)$$

An achievable MM-SPIR scheme is a scheme that satisfies the MM-SPIR reliability constraint (15), the user privacy constraint (16), and the database privacy constraint (17). Following the definition of the sum retrieval rate of $W_P$ in [20], we define the sum retrieval rate of MM-SPIR as,

$$R_{MM-SPIR} = \frac{H(W_P)}{H(A_n^{[P]}|P, Q_{1:N}^{[P]}, F)} = \frac{PL}{N} \quad (18)$$

The sum capacity of MM-SPIR, $C_{MM-SPIR}$, is the supremum of the sum retrieval rates over all achievable schemes.

B. MM-SPIR: Main Results

**Theorem 2** The MM-SPIR capacity for $N \geq 2$, $K \geq 2$, and $P \leq K$, is given by,

$$C_{MM-SPIR} = \begin{cases} 1, & P = K \\ 1 - \frac{1}{N}, & 1 \leq P \leq K - 1, \quad H(S) \geq \frac{PL}{N-1} \\ 0, & \text{otherwise} \end{cases} \quad (19)$$

We note that the result implies that the capacity of MM-SPIR is exactly the same as the capacity of SM-SPIR [7]. Hence, there is no gain from joint retrieval in comparison to successive single-message SPIR [7]. This in contrast to the gain in MM-PIR [20] in comparison to successive single-message PIR [6]. MM-SPIR capacity expression in Theorem 2 inherits all of the structural remarks from [7]. Furthermore, for the extreme case of $P = K$, the SPIR capacity is 1 without using any common randomness. This is due to the fact that the user privacy and the database privacy constraints are trivially satisfied, and hence the user can simply download all of the messages from one of the databases without using any common randomness.

C. MM-SPIR: Converse Proof

To prove the converse of Theorem 2, we first need the following lemmas. Lemmas 1 and 2 are direct extensions to [7, Lemmas 1 and 2] to the setting of MM-SPIR.
Lemma 1 (Symmetry) \( \forall n, \forall P_1 \neq P_2 \text{ s.t. } |P_1| = |P_2| \)

\[
H(A_n^{[P_1]}|W_{P_1}, Q_n^{[P_1]}) = H(A_n^{[P_2]}|W_{P_1}, Q_n^{[P_2]})
\]

(20)

\[
H(A_n^{[P_1]}|Q_n^{[P_1]}) = H(A_n^{[P_2]}|Q_n^{[P_2]})
\]

(21)

Lemma 2 (Effect of conditioning on user’s randomness)

\[
H(A_n^{[P]}|W_P, F, Q_n^{[P]}) = H(A_n^{[P]}|W_P, Q_n^{[P]}), \forall n, \forall P
\]

(22)

Next, we need Lemma 3, which is an existence proof for index sets with specific properties. This technical lemma is needed in the proofs of upcoming two lemmas, Lemma 4 and Lemma 5. First, we give the definitions of relevant index sets \( P_a, P_b, P_c, P_d \), and an element \( i_m \). Given \( P_1 \) and \( P_2 \), we divide \( P_1 \) into two disjoint partitions \( P_a \) and \( P_b \) (i.e., \( P_a \cap P_b = P_1 \) and \( P_a \cap P_b = \emptyset \)), where \( P_a \subseteq P_2 \) (i.e., \( P_1 \cap P_2 = P_a \)), \( P_b \subseteq P_2 \). Suppose \( |P_a| = M \in [1 : P - 1] \). Note that since \( P_1 \neq P_2 \), we cannot have \( M = P \). We assume that \( P_a = \{i_1, \ldots, i_m\} \) which consists of exactly the first \( m \) elements in the index set \( P_a \). Let \( i_m \) be the last element from the index set \( P_a \). We obtain a new index set \( P_d = \{i_1, \ldots, i_{m-1}\} \) after removing this element. That means \( P_c = P_d \cup \{i_m\} \).

Lemma 3 For \( k \geq 3, 1 \leq p \leq k - 1 \), given index sets \( P_1, P_2 \) such that \( |P_1| = p \) for \( i = 1, 2 \) and \( P_1 \neq P_2 \), we can construct an index set \( P_3 \) such that,

i. \( P_3 \neq P_1 \) and \( P_3 \neq P_2 \),

ii. \( |P_3| = p \), and

iii. \( P_3 \) includes \( P_b \cup P_d \) but does not include the common element \( i_m \) in \( P_1 \cap P_2 \).

Next, we need the following lemma. Lemma 4 states that revealing any individual answer given the messages \( (W_{P_1}, W_{P_2}) \) does not leak any information about the message \( W_{i_m} \).

Lemma 4 (Message leakage within any answer string)

When \( 1 \leq p \leq k - 1 \) and \( M \geq 1 \), for arbitrary \( m \in [1 : M] \),

\[
H(W_{i_m}|W_{P_1}, W_{P_2}, A_n^{[P_2]}, Q_n^{[P_2]}) = H(W_{i_m}|W_{P_1}, W_{P_2}, Q_n^{[P_2]})
\]

(23)

Finally, we prove that conditioning on an undesired message set does not decrease the uncertainty on any answer string.

Lemma 5 (Conditioning on an undesired message set)

\[
H(A_n^{[P]}|W_{P_1}, Q_n^{[P]}) = H(A_n^{[P]}|Q_n^{[P]}), \forall n, \forall P_2 \text{ s.t. } P_1 \neq P_2, |P_1| = |P_2|
\]

(24)

The proof for \( R \leq C_{MM-SPIR} \):

\[
PL = H(W_{P_1})
\]

(25)

\[
\stackrel{(11)}{=} H(W_{P_1}|F)
\]

(26)

\[
\stackrel{(15)}{=} H(W_{P_1}|F) - H(W_{P_1}|A_{1:N}^{[P]}|F)
\]

(27)

\[
= I(W_{P_1}; A_{1:N}^{[P]}|F)
\]

(28)

\[
= H(A_{1:N}^{[P]}|F) - H(A_{1:N}^{[P]}|W_{P_1}, F)
\]

(29)

\[
\stackrel{(12)}{=} H(A_{1:N}^{[P]}|F) - H(A_{1:N}^{[P]}|W_{P_1}, F, Q_n^{[P]})
\]

(30)

\[
\leq H(A_{1:N}^{[P]}|F) - H(A_{1:N}^{[P]}|W_{P_1}, F, Q_n^{[P]})
\]

(31)

\[
\stackrel{(20)}{=} H(A_{1:N}^{[P]}|F) - H(A_{1:N}^{[P]}|Q_n^{[P]})
\]

(32)

\[
\leq H(A_{1:N}^{[P]}|F) - H(A_{1:N}^{[P]}|Q_n^{[P]})
\]

(33)

\[
\leq H(A_{1:N}^{[P]}|F) - H(A_{1:N}^{[P]}|Q_n^{[P]})
\]

(34)

\[
\leq H(A_{1:N}^{[P]}|F) - H(A_{1:N}^{[P]}|Q_n^{[P]})
\]

(35)

\[
\leq H(A_{1:N}^{[P]}|F) - H(A_{1:N}^{[P]}|Q_n^{[P]})
\]

(36)

\[
\leq H(A_{1:N}^{[P]}|F) - H(A_{1:N}^{[P]}|F)
\]

(37)

By summing (37) up for all \( n \in [1 : N] \) and letting \( P \) denote the general desired index set, we obtain,

\[
NPL \leq NH(A_{1:N}^{[P]}|F) - \sum_{n=1}^{N} H(A_{1:N}^{[P]}|F)
\]

(38)

\[
\leq NH(A_{1:N}^{[P]}|F) - H(A_{1:N}^{[P]}|F)
\]

(39)

\[
= (N - 1)H(A_{1:N}^{[P]}|F)
\]

(40)

\[
\leq (N - 1) \sum_{n=1}^{N} H(A_{1:N}^{[P]}|F)
\]

(41)

\[
\leq (N - 1) \sum_{n=1}^{N} H(A_{1:N}^{[P]})
\]

(42)

which leads to the desired converse result on the retrieval rate,

\[
R = \frac{PL}{\sum_{n=1}^{N} H(A_{1:N}^{[P]})} \leq \frac{N - 1}{N} = 1 - \frac{1}{N}
\]

(43)

The proof for \( H(S) \geq \frac{PL}{N} \):

\[
0 \stackrel{(11)}{=} I(W_{P_1}; A_{1:N}^{[P]}|Q_{1:N}^{[P]}, F)
\]

(44)

\[
\geq I(W_{P_1}; A_{1:N}^{[P]}|F)
\]

(45)

\[
= I(W_{P_1}; A_{1:N}^{[P]}|W_{P_1}, F)
\]

(46)

\[
\stackrel{(15)}{=} I(W_{P_1}; A_{1:N}^{[P]}|W_{P_1}, F)
\]

(47)

\[
\geq I(W_{P_1}; A_{1:N}^{[P]}|W_{P_1}, F)
\]

(48)

\[
= H(A_{1:N}^{[P]}|W_{P_1}, F) - H(A_{1:N}^{[P]}|W_{1:K}, F)
\]

(49)

\[
\leq H(A_{1:N}^{[P]}|W_{P_1}, F) - H(A_{1:N}^{[P]}|W_{1:K}, F)
\]

(50)

\[
= H(A_{1:N}^{[P]}|W_{P_1}, F) - I(S; A_{1:N}^{[P]}|W_{1:K}, F)
\]

(51)

\[
= H(A_{1:N}^{[P]}|W_{P_1}, F) - H(S|W_{1:K}, F)
\]

(52)
where (56) follows from the steps between (32)-(35) by applying Lemma 1, 2 and 5 again.

By summing (56) up for all \( n \in [1 : N] \) and letting \( P \) denote the general desired index set again, we obtain,

\[
0 \geq \sum_{n=1}^{N} H(A_n^{[P]}|Q_n^{[P]}) - NH(S) \quad (57)
\]

\[
\geq H(A_n^{[P]}|Q_n^{[P]}) - NH(S) \quad (58)
\]

\[
\geq H(A_n^{[P]}|Q_n^{[P]}, \mathcal{F}) - NH(S) \quad (59)
\]

\[
= H(A_n^{[P]}|Q_n^{[P]}) - H(S) \quad (60)
\]

\[
\geq \frac{N}{N-1} PL - NH(S) \quad (61)
\]

where (60) follows from (12) and (61) follows from (40), which leads to a lower bound for the minimal required entropy of common randomness \( S \),

\[
H(S) \geq \frac{PL}{N-1} \quad (62)
\]

D. MM-SPIR: Achievability Proof

Since the MM-SPIR capacity is the same as the SM-SPIR capacity, and the required common randomness is \( P \) times the required common randomness for SM-SPIR, we can use the achievable scheme in [7] successively \( P \) times in a row (by utilizing independent common randomness each time) to achieve the MM-SPIR capacity. Although the query structure for the capacity-achieving scheme for SPIR in [7] is quite simple, it is fundamentally different than the query structure for the capacity-achieving scheme for PIR in [6]. This means that user/databases should execute different query structures for different database privacy levels. In this paper, by combining ideas for achievability from [20] and [13], we propose an alternative capacity-achieving scheme for MM-SPIR for any \( P \). Our achievable scheme enables us to switch between MM-PIR and MM-SPIR seamlessly, and therefore support different database privacy levels, as the basic query structures are similar.

For convenience, we use the \( k \)-sum notation in [6], [20]. A \( k \)-sum is a sum of \( k \) symbols from \( k \) different messages. Thus, a \( k \)-sum symbol appears only in round \( k \). We denote the number of stages in round \( k \) by \( \alpha_k \), which was originally introduced in [20]. In addition, we use \( \nu \) to denote the number of repetitions of the scheme in [20] we need before we start assigning common randomness symbols.

Our achievability scheme is primarily based on the one in [20], with the addition of downloading and/or mixing common randomness variables into symbol downloads appropriately. We note that, here we extend the near-optimal algorithm in [20], which was originally proposed for \( P \leq \frac{K}{2} \), to the case of \( P \geq \frac{K}{2} \), and therefore, use it for all \( 1 \leq P \leq K-1 \). Our achievability scheme comprises the following steps:

1) Initial MM-PIR Query Generation: Generate an initial query table strictly following the near-optimal procedure in [20] for arbitrary \( K, P \) and \( N \).

2) Repetition: Repeat Step 1 for a total of \( \nu \) times. The purpose of the repetition is to i) get an integer number of common randomness generated at each database by a symmetric algorithm, and ii) get equal number of symbols downloaded from each desired message. Let \( \nu_0 \) be the smallest integer such that \( (N-1)^{K-P}\nu_0 \) is an integer. Similarly, for \( 1 \leq k \leq \min\{P,K-P\} \), let \( \nu_k \) be the smallest integer such that \( \frac{(N-1)^{K-P}\nu_k}{N^{\nu_0}} \) is an integer. Choose \( \nu \) as the lowest common multiple of all \( \nu_k \).

3) Common Randomness Assignment:

   a) In round 1, assign \( \frac{\nu P \alpha_1}{N} \) independent common randomness symbols to each database, and download them. At each database, mix every 1-sum symbol containing a desired message symbol with an arbitrary common randomness already downloaded from another database, making sure that every 1-sum symbol at each database is mixed with a different common randomness symbol. Mix all other 1-sum symbols not containing a desired symbol with a new common randomness symbol which is not downloaded by the user.

   b) In round \( k \) (\( k \geq 2 \)), assign \( \frac{\nu_k P \alpha_k}{N} \) independent common randomness symbols to each database, and download them. At each database: Mix every \( k \)-sum symbol containing only desired message symbols with an arbitrary common randomness symbol already downloaded from another database. Mix every \( k \)-sum symbol containing \( p \) desired message symbols (\( 1 \leq p \leq k-1 \)) with the common randomness symbol from the \((k-p)\)-sum symbol having the same \( k-p \) undesired message symbols downloaded at any other database. Mix every \( k \)-sum symbol not containing any desired message symbols with a new common randomness symbol which is not downloaded by the user.

   c) Repeat Step 3b until \( k \) reaches \( K \). Note that if \( \alpha_k = 0 \), nothing is done.

VI. PUTTING EVERYTHING TOGETHER

Finally, we map our MM-SPIR result in Theorem 2 back to the PSI problem to obtain Theorem 1. Recall that, in the PSI problem, by generating the sets \( P_1 \) and \( P_2 \) by i.i.d. drawing the elements, we obtain i.i.d. messages in the MM-SPIR problem. Further, by choosing \( q_i = \frac{1}{2} \), we obtain uniformly distributed messages, with message size \( L = 1 \). Therefore, the PSI problem is equivalent to an MM-SPIR problem with \( L = 1 \).

We extend our MM-SPIR results for a finite message size \( L \) straightforwardly, see [52, Theorem 3]. Now, using this with \( L = 1 \), we obtain the result of this paper in Theorem 1.
REFERENCES


