Symmetric Private Information Retrieval with User-Side Common Randomness

Zhusheng Wang  
Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742
zhusheng@umd.edu

Sennur Ulukus
zhusheng@umd.edu

Abstract—We consider the problem of symmetric private information retrieval (SPIR) with user-side common randomness. In SPIR, a user retrieves a message out of $K$ messages from $N$ non-colluding and replicated databases in such a way that no single database knows the retrieved message index (user privacy), and the user gets to know nothing further than the retrieved message (database privacy). SPIR has a capacity smaller than the PIR capacity which requires only user privacy, is infeasible in the case of a single database, and requires shared common randomness among the databases. We introduce a new variant in the case of a single database, and requires shared common randomness, which is unknown to the databases. We determine the exact capacity region of the triple $(d, \rho_S, \rho_U)$, where $d$ is the download cost, $\rho_S$ is the amount of shared database (server) common randomness, and $\rho_U$ is the amount of available user-side common randomness. We show that with a suitable amount of $\rho_U$, this new SPIR achieves the capacity of conventional PIR. As a corollary, single-database SPIR becomes feasible. Further, the presence of user-side $\rho_U$ reduces the amount of required server-side $\rho_S$.

I. INTRODUCTION

Private information retrieval (PIR) is a fundamental problem where a user downloads a message out of $K$ possible messages stored in $N$ non-colluding and replicated databases in such a way that no single database can know which message the user has downloaded [1]. This privacy requirement is referred to as user privacy. Symmetric PIR (SPIR) is an extended version of PIR where while downloading its desired message the user learns nothing about the remaining messages stored in the databases [2]. This is referred to as database privacy. While PIR can be achieved with no shared common randomness, it is well-known that information-theoretic SPIR is possible only when the databases share a certain minimum amount of common randomness that is unknown to the user.

The information-theoretic capacity of PIR and SPIR have been found in [3], [4] as $C_{\text{PIR}} = \frac{1}{1-\frac{1}{K}}$ and $C_{\text{SPIR}} = 1 - \frac{1}{N}$. First, $C_{\text{SPIR}}$ is smaller than $C_{\text{PIR}}$, since SPIR is a more constrained problem than PIR, as it requires not only user privacy but also database privacy. Second, single-database SPIR is infeasible as $C_{\text{SPIR}} = 0$ for $N = 1$, while single database PIR is feasible as $C_{\text{PIR}} = \frac{1}{K}$ for $N = 1$. Our goal in this paper is two-fold: To explore ways to increase SPIR capacity to the level of PIR capacity, and perhaps more importantly, to make single-database SPIR feasible.

Our motivation to focus on SPIR comes from constantly growing importance of privacy, not only the privacy of the retrieving user, but also the privacy of the databases, as the stored information in the databases may belong to other users. In addition, recent papers showed that other important privacy primitives, such as private set intersection (PSI) [5], [6], can be recast as versions of the SPIR problem, e.g., multi-message PSI. Thus, here, we investigate ways to increase the SPIR capacity. Further, in practical applications, enforcing non-collusion could be difficult, as in some cases, all databases may naturally belong to the same entity, e.g., in various multi-party secure computation problems [5]–[12]. If all databases collude or belong to the same entity, the system essentially becomes a single-database system [13]. The single-database PIR problem has been studied under extended conditions, e.g., side-information [14]–[17]. Here, we investigate single-database SPIR under extended conditions, with the goal of making it feasible. Other important variations of PIR and SPIR problem have also been investigated; see e.g., [18]–[51].

In this paper, we introduce SPIR with user-side common randomness, which solves the above two issues. In this model, the user obtains a part of the common randomness shared by the databases. The databases know the size of the user-side common randomness, but they do not know what the user possesses exactly. One way to implement this is for the user to fetch a part of the common randomness from the databases uniformly randomly, i.e., without the user knowing what it will get and without the databases knowing what it got, except for its cardinality. That is, all subsets of a certain size are equally likely to be obtained by the user. Another practical implementation could be for an external helper to distribute common randomness to the user and the databases randomly.

For database-side (server-side) common randomness of amount $\rho_S$ and user-side common randomness of amount $\rho_U$, we determine the exact capacity region of the triple $(d, \rho_S, \rho_U)$, where $d$ is the download cost which is the inverse of the capacity. We show that with a suitable $\rho_U$, SPIR capacity becomes equal to the conventional PIR capacity. For the single-database case, since the conventional PIR capacity is $\frac{1}{K}$, this implies that single-database SPIR with user-side common randomness is feasible. In addition, the presence of user-side $\rho_U$ reduces the amount of required server-side $\rho_S$.

II. PROBLEM FORMULATION

We consider a system of $N \geq 1$ non-colluding databases each storing the same set of $K \geq 2$ i.i.d. messages each of
which consisting of $L$ i.i.d. symbols uniformly selected from a sufficiently large finite field $F_q$, i.e.,

$$H(W_k) = L, \quad k \in [K]$$

(1)

$$H(W_{1:K}) = H(W_1) + \cdots + H(W_K) = KL$$

(2)

As in [4], we use a random variable $F$ to denote the randomness in the retrieval strategy implemented by the user. Due to the user privacy constraint, the realization of $F$ is only known to the user, and is unknown to any of the databases. Due to the database privacy constraint, databases need to share some amount of common randomness $R_{S};$ we will call this server-side common randomness. Before the retrieval process starts, the user obtains a partial knowledge of $R_{S}.$ We denote it by $R_{U},$ and call it user-side common randomness. $R_{U}$ is a subset of $R_{S}.$ The user-side common randomness $R_{U}$ is unknown to the databases, i.e., it is only known to be equally distributed among all subsets of $R_{S}$ with cardinality $|R_{U}|.$

The message set $W_{1:K}$ stored in the databases is independent of the desired message index $k,$ retrieval strategy randomness at the user $F$ and all the common randomness,

$$I(W_{1:K}; k, F, R_{S}, R_{U}) = 0, \quad \forall k, \forall R_{U}$$

(3)

During the query generation stage, the user has no access to the message set $W_{1:K}$ in the databases and the common randomness difference $R_{S} \setminus R_{U},$

$$I(Q_{1:N}^{[k, R_{U}]}, W_{1:K}, R_{S} \setminus R_{U}) = 0, \quad \forall k, \forall R_{U}$$

(4)

Using the desired message index and the user-side common randomness, the user generates a query for each database according to the retrieval strategy randomness $F.$ Hence, the queries $Q_{n}^{[k, R_{U}]}, n \in [N]$ are deterministic functions of $F,$

$$H(Q_{1}^{[k, R_{U}]}, Q_{2}^{[k, R_{U}]}, \ldots, Q_{N}^{[k, R_{U}]} | F) = 0 \quad \forall k, \forall R_{U}$$

(5)

After receiving a query from the user, each database generates a truthful answer based on the stored message set and the server-side common randomness,

$$H(A_{n}^{[k, R_{U}]} | Q_{n}^{[k, R_{U}]}, W_{1:K}, R_{S}) = 0, \quad \forall n, \forall k, \forall R_{U}$$

(6)

After collecting all $N$ answers from the databases, the user should be able to decode the desired messages $W_{k}$ reliably,

$$H(W_{k} | F, A_{1:N}^{[k, R_{U}]}, R_{U}) = 0, \quad \forall k, \forall R_{U}$$

(7)

Due to the user privacy constraint, the query generated to retrieve the desired message should be statistically indistinguishable from other queries. Specifically, for all $k, k', n$, and all user-side common randomness $R_{U},$ there exists some $R'_{U}$ with $H(R'_{U}) = H(R_{U})$ such that,

$$H(W_{k} | F, A_{1:N}^{[k, R'_{U}]}, R_{U}) = 0, \quad \forall k, \forall R_{U}$$

[user privacy] $P(Q_{n}^{[k, R'_{U}]} = q) = P(Q_{n}^{[k', R'_{U}]} = q)$

(8)

Due to the database privacy constraint, the user should learn nothing about $W_{k}$ which is the complement of $W_{k},$ i.e., $W_{k} = \{W_{1}, \ldots, W_{k-1}, W_{k+1}, \ldots, W_{K}\},$

[database privacy] $I(W_{k}; F, A_{1:N}^{[k, R'_{U}]}, R_{U}) = 0$

(9)

Again due to the database privacy, the user should not gain any knowledge about the remaining common randomness in the server even after retrieving the desired message,

$$I(R_{S} \setminus R_{U}; F, A_{1:N}^{[k, R'_{U}]}, W_{k}, R_{U}) = 0$$

(10)

An achievable SPIR scheme is a scheme that satisfies the reliability constraint (7), the user privacy constraint (8) and the database privacy constraint (10). As usual, the efficiency of the scheme is measured in terms of the maximal number of downloaded bits by the user from all the databases, denoted by $D.$ We define the normalized download cost $d,$ the normalized server-side common randomness $\rho_{S},$ and the normalized user-side common randomness as $\rho_{U}$ as,

$$d = \frac{D}{L}, \quad \rho_{S} = \frac{H(R_{S})}{L}, \quad \rho_{U} = \frac{H(R_{U})}{L}$$

(12)

where $L$ is the message length. Our goal in this paper is to determine the largest region for simultaneously achievable triples $(d, \rho_{S}, \rho_{U})$ over all valid retrieval schemes.

III. MAIN RESULTS

We state the main result of our paper in the following theorem which is the capacity region for the triple $(d, \rho_{S}, \rho_{U}).$

Theorem 1 With user-side common randomness, the multi-database SPIR capacity region for $N \geq 2$ and $K \geq 2$ is

$$d \geq 1 + \frac{1}{N} + \frac{1}{N^2} + \cdots + \frac{1}{N^{K-1}}$$

(13)

$$\rho_{S} - \rho_{U} \geq \frac{1}{N} + \frac{1}{N^2} + \cdots + \frac{1}{N^{K-1}}$$

(14)

$$\frac{N-1}{N} - d + \rho_{U} \geq 1$$

(15)

$$\frac{N}{N-1} \rho_{U} + N \rho_{S} \geq \frac{N}{N-1}$$

(16)

Remark 1 The right hand side of (13) is the optimum normalized download cost of classical PIR, $d_{PIR}$ [3]. Thus, (13) states that $d \geq d_{PIR}.$ When $\rho_{U} = 0,$ i.e., when there is no user-side common randomness, (15) becomes $d \geq d_{SPIR},$ where $d_{SPIR} = \frac{N}{N-1}$ is the optimum normalized download cost of classical SPIR [4]. Note that $d_{SPIR} > d_{PIR}$ for all $N.$ Therefore, when $\rho_{U} = 0$, (15) is binding, (13) is loose, and we have $d = d_{SPIR}.$

Theorem 1 implies that with appropriate $\rho_{U},$ e.g., with $\rho_{U} = \frac{1}{N}$, both (13) and (15) can be made binding, at which time the new SPIR download cost achieves $d = d_{PIR}.$

Remark 2 When $\rho_{U} = 0$, Theorem 1 reduces to the capacity of classical SPIR [4], as in this case, (15) gives $d \geq \frac{N}{N-1},$ (16) gives $\rho_{S} \geq \frac{1}{N-1},$ and (13) and (14) are non-binding.
Remark 3 The gap between $p_S$ and $p_U$ must be no smaller than a specific value as a function of $N$ and $K$ as given on the right hand side of (14). This comes from the server privacy constraint, where part of the common randomness, i.e., $R_S \setminus R_U$, is utilized to hide the undesired messages.

Remark 4 From (16), we observe that the existence of user-side common randomness can help reduce the required amount of server-side common randomness. For instance, for $N = 2$ databases and $K = 2$ messages, classical SPIR optimum download cost $d = d_{\text{SPIR}} = 2$ is achieved by $p_S = 1$ [4]. In Theorem 1, $d = 2$ can be achieved by $p_S = \frac{3}{4}$ with $p_U = \frac{1}{4}$.

Corollary 1 With user-side common randomness, the single-database SPIR capacity region for $N = 1$ and $K \geq 2$ is

\begin{equation}
q_S - p_U \geq K - 1
\end{equation}

(18)

Remark 5 It is well-known that, for $N = 1$, classical SPIR is not feasible [4]. With user-side common randomness, single-database SPIR becomes feasible.

Remark 6 The optimal normalized download cost for single-database PIR is $d = K$ [3], [14], which is achieved by downloading all messages from the server. One of the difficulties of single-database SPIR is that downloading all messages is not a valid SPIR scheme. Corollary 1 shows that single-database PIR capacity can be achieved for single-database SPIR by means of user-side common randomness.

Remark 7 The first two terms in Corollary 1 follow from the first two terms in Theorem 1. The third term in Corollary 1 follows from the last two terms in Theorem 1 by multiplying both sides of the fourth term in Theorem 1 by $N - 1$.

Remark 8 Like multi-database SPIR, in the single-database SPIR as well, the gap between $p_S$ and $p_U$ must be no smaller than a specific value as a function of $N$ as given in (18) to avoid information leakage on undesired messages.

IV. MOTIVATING EXAMPLE

Example 1 We consider a single-database case $N = 1, K = 3$ and $L = 1$. We use $W_1$, $W_2$, and $W_3$ to denote the three messages. Our new achievable scheme is given in Table I.

The reliability constraint follows from the fact that the user can always decode the desired message by using its own common randomness. The server privacy constraint follows from the fact that the undesired messages are always mixed with unknown common randomness. For the user-privacy constraint, we have for all $k, k' \in [3], k \neq k'$ and a randomly selected $R_U \in \{S_1, S_2, S_3\}$ under a uniform distribution, there exists another different $R'_U \in \{S_1, S_2, S_3\}$ such that,

\begin{equation}
P(Q^{[k,R_U]} = q) = P(Q^{[k',R'_U]} = q) = \frac{1}{3}
\end{equation}

TABLE I: The query table for the case $N = 1, K = 3$.

<table>
<thead>
<tr>
<th>$R_U$</th>
<th>Desired message: $W_1$</th>
<th>Desired message: $W_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>$W_1 + S_1$</td>
<td>$W_2 + S_1$</td>
</tr>
<tr>
<td>$W_2 + S_2$</td>
<td>$W_3 + S_2$</td>
<td>$W_1 + S_2$</td>
</tr>
<tr>
<td>$W_3 + S_3$</td>
<td>$W_1 + S_3$</td>
<td>$W_2 + S_3$</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$W_1 + S_2$</td>
<td>$W_2 + S_2$</td>
</tr>
<tr>
<td>$W_2 + S_3$</td>
<td>$W_3 + S_3$</td>
<td>$W_1 + S_3$</td>
</tr>
<tr>
<td>$W_3 + S_1$</td>
<td>$W_1 + S_3$</td>
<td>$W_2 + S_1$</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$W_1 + S_3$</td>
<td>$W_2 + S_3$</td>
</tr>
<tr>
<td>$W_2 + S_1$</td>
<td>$W_3 + S_1$</td>
<td>$W_1 + S_1$</td>
</tr>
<tr>
<td>$W_3 + S_2$</td>
<td>$W_1 + S_2$</td>
<td>$W_2 + S_2$</td>
</tr>
</tbody>
</table>

where $q \in \{[W_1 + S_1, W_2 + S_2, W_3 + S_3], [W_1 + S_2, W_2 + S_3, W_3 + S_1], [W_1 + S_3, W_2 + S_1, W_3 + S_2]\}$. Specifically from the point of view of the server, the same set of queries can be invoked for any desired message $W_i$, $i \in [3]$ with the same probability distribution. This scheme achieves $d = 3$, $p_U = 1$ and $p_S = 3$, which exactly matches the boundary of the SPIR capacity region for $N = 1$ and $K = 3$ in Corollary 1.

Example 2 We consider a multi-database case $N = 2, K = 2$ and $L = 4$. We use $[a_1, a_2, a_3, a_4]$ as a random uniform permutation of the symbols in the first message $W_1$, and independently, $[b_1, b_2, b_3, b_4]$ as another one for $W_2$. Due to message index permutations, each set of queries represents one of $4! \cdot 3 = 288$ different possible permutations. We have two different sets for each $R_U$ because of necessary common randomness permutations. Our new achievable scheme for one random message index selection is given in Table II.

<table>
<thead>
<tr>
<th>$R_U$</th>
<th>Desired message: $W_1$</th>
<th>Desired message: $W_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>$a_1 + S_1$</td>
<td>$b_1 + S_1$</td>
</tr>
<tr>
<td>$a_2 + S_1$</td>
<td>$a_1 + S_2$</td>
<td>$a_2 + S_1$</td>
</tr>
<tr>
<td>$a_3 + b_2 + S_4$</td>
<td>$a_4 + b_1 + S_2$</td>
<td>$b_1 + S_3$</td>
</tr>
<tr>
<td>$b_3 + a_2 + S_3$</td>
<td>$a_1 + S_3$</td>
<td>$b_2 + S_2$</td>
</tr>
<tr>
<td>$a_1 + S_1$</td>
<td>$a_2 + S_1$</td>
<td>$b_1 + S_3$</td>
</tr>
<tr>
<td>$b_1 + S_3$</td>
<td>$a_2 + S_3$</td>
<td>$b_1 + S_2$</td>
</tr>
<tr>
<td>$a_3 + b_2 + S_4$</td>
<td>$a_4 + b_1 + S_2$</td>
<td>$b_1 + S_3$</td>
</tr>
<tr>
<td>$a_1 + S_1$</td>
<td>$a_2 + S_1$</td>
<td>$b_1 + S_3$</td>
</tr>
<tr>
<td>$a_2 + S_1$</td>
<td>$a_3 + b_2 + S_4$</td>
<td>$b_1 + S_2$</td>
</tr>
<tr>
<td>$a_3 + b_2 + S_4$</td>
<td>$a_4 + b_1 + S_2$</td>
<td>$b_1 + S_3$</td>
</tr>
<tr>
<td>$a_1 + S_2$</td>
<td>$a_2 + S_1$</td>
<td>$b_1 + S_3$</td>
</tr>
<tr>
<td>$b_1 + S_3$</td>
<td>$a_2 + S_3$</td>
<td>$b_1 + S_2$</td>
</tr>
<tr>
<td>$a_3 + b_2 + S_4$</td>
<td>$a_4 + b_1 + S_2$</td>
<td>$b_1 + S_3$</td>
</tr>
<tr>
<td>$a_1 + S_3$</td>
<td>$a_2 + S_1$</td>
<td>$b_1 + S_3$</td>
</tr>
<tr>
<td>$a_2 + S_1$</td>
<td>$a_3 + b_2 + S_4$</td>
<td>$b_1 + S_2$</td>
</tr>
<tr>
<td>$a_3 + b_2 + S_4$</td>
<td>$a_4 + b_1 + S_2$</td>
<td>$b_1 + S_3$</td>
</tr>
</tbody>
</table>

TABLE II: The query table for the case $N = 2, K = 2$.

Verification that this proposed scheme achieves the user privacy and the database privacy constraints is similar to the one in Example 1. This scheme achieves $d = 1.5$, $p_U = 0.25$ and $p_S = 0.75$. This is a corner point of the capacity region.
in Theorem 1 where all inequalities are satisfied with equality. The other corner point when $\rho_U = 0$ is achieved by the classical SPIR scheme in [4]. Any point on the line segment joining these two points can be achieved by time-sharing between these two schemes. Any other remaining point in Theorem 1 can be achieved by adding extra randomness in the user- and server-side simultaneously, or by increasing the server-side common randomness and the download cost.

V. CONVERSE PROOF

We provide a sketch of the converse proof of Theorem 1 here. The four inequalities in Theorem 1 are proved in Lemmas 3, 4, 9 and 10 below. Towards proving these four lemmas, we need Lemmas 1-2 and Lemmas 5-8 below. We note that Lemmas 1-2 extend [3, Lemmas 5-6], and Lemmas 5-8 extend [4, Eqns. (26), (27), (30), (39)]. These extensions are needed because we have two additional sets of random variables in our system model: $\mathcal{R}_S$ and $\mathcal{R}_U$ with respect to techniques in [3], and $\mathcal{R}_U$ with respect to techniques in [4].

Lemma 1

$$I(W_{2;K};Q^{[1,R_U]}_1, A^{[1,R_U]}_1, \mathcal{R}_S|W_1) \leq D - L$$ (21)

Lemma 2

$$I(W_{k;K};Q^{[k-1,R_U]}_1, A^{[k-1,R_U]}_1, \mathcal{R}_S|W_{1:k-1}) \geq \frac{1}{N} I(W_{k+1:K};Q^{[k,R_U]}_1, A^{[k,R_U]}_1, \mathcal{R}_S|W_{1:k}) + \frac{L}{N}$$ (22)

Lemma 3 (Minimal download cost $d$)

$$d \geq 1 + \frac{1}{N} + \frac{1}{N^2} + \cdots + \frac{1}{N^{K-1}}$$ (23)

Proof: Following steps similar to [3, Eqns. (62)-(67)] for Lemma 2, we obtain

$$I(W_{2;K};Q^{[1,R_U]}_1, A^{[1,R_U]}_1, \mathcal{R}_S|W_1) \geq \left( \frac{1}{N} + \frac{1}{N^2} + \cdots + \frac{1}{N^{K-1}} \right) L$$ (24)

Combining the upper bound in Lemma 1 and the lower bound in (24) completes the proof. ■

Lemma 4 (Minimal difference between $\rho_S$ and $\rho_U$)

$$\rho_S - \rho_U \geq \frac{1}{N} + \frac{1}{N^2} + \cdots + \frac{1}{N^{K-1}}$$ (25)

Proof: From (24), we have the following relation,

$$H(W_{2;K}|Q^{[1,R_U]}_1, A^{[1,R_U]}_1, W_1, \mathcal{R}_S) \leq (K-1)L - \left( \frac{1}{N} + \frac{1}{N^2} + \cdots + \frac{1}{N^{K-1}} \right) L$$ (26)

Next, we have the following upper bound,

$$I(W_{2;K};\mathcal{R}_S \setminus \mathcal{R}_U|Q^{[1,R_U]}_1, A^{[1,R_U]}_1, W_1, \mathcal{R}_U) \leq H(\mathcal{R}_S \setminus \mathcal{R}_U|Q^{[1,R_U]}_1, A^{[1,R_U]}_1, W_1, \mathcal{R}_U)$$ (27)

and the following lower bound,

$$I(W_{2;K};\mathcal{R}_S \setminus \mathcal{R}_U|Q^{[1,R_U]}_1, A^{[1,R_U]}_1, W_1, \mathcal{R}_U) \geq \frac{1}{N} \left( \frac{1}{N} + \frac{1}{N^2} + \cdots + \frac{1}{N^{K-1}} \right) L$$ (28)

Combining (28) and (30) yields the desired result. ■

Lemma 5 For any $k \neq k'$,

$$H(A^{[k,R_U]}_n|Q^{[k,R_U]}_n, W_k, \mathcal{R}_U) \geq H(A^{[k',R_U]}_n|Q^{[k,R_U]}_n, W_k, \mathcal{R}_U') - H(\mathcal{R}_U')$$ (31)

Lemma 6

$$H(A^{[k,R_U]}_n|Q^{[k,R_U]}_n, \mathcal{R}_U) = H(A^{[k,R_U]}_n|Q^{[k,R_U]}_n, W_k, \mathcal{R}_U)$$ (32)

Lemma 7

$$H(A^{[k,R_U]}_n|Q^{[k,R_U]}_n, W_k, \mathcal{R}_U) = H(A^{[k,R_U]}_n|Q^{[k,R_U]}_n, W_k, \mathcal{R}_U')$$ (33)

Lemma 8 For any $k \neq k'$,

$$H(A^{[k',R_U]}_n|Q^{[k',R_U]}_n, \mathcal{R}_U') = H(A^{[k',R_U]}_n|Q^{[k',R_U]}_n, W_k, \mathcal{R}_U')$$ (34)

Lemma 9 (Minimal bound for $d$ and $\rho_U$)

$$\frac{N-1}{N} d + \rho_U \geq 1$$ (35)

Proof: Starting from the message length assumption (1),

$$L = H(W_k) \stackrel{(3)}{=} H(W_k|\mathcal{F}, \mathcal{R}_U)$$ (36)

$$= H(W_k|\mathcal{F}) - H(W_k|\mathcal{F}, A^{[k,R_U]}_1, \mathcal{R}_U)$$ (37)

$$= I(W_k;A^{[k,R_U]}_1|\mathcal{F}, \mathcal{R}_U)$$ (38)

$$\geq H(A^{[k,R_U]}_1|\mathcal{F}, \mathcal{R}_U') - H(A^{[k,R_U]}_1|\mathcal{F}, W_k, \mathcal{R}_U')$$ (39)

$$\geq H(A^{[k,R_U]}_1|\mathcal{F}, \mathcal{R}_U') - H(A^{[k,R_U]}_1|\mathcal{R}_U')$$ (40)

$$\quad \leq H(A^{[k,R_U]}_1|\mathcal{F}, \mathcal{R}_U') - H(A^{[k,R_U]}_1|Q^{[k,R_U]}_n, W_k, \mathcal{R}_U')$$ (41)

$$\quad \leq H(A^{[k,R_U]}_1|\mathcal{F}, \mathcal{R}_U') - H(A^{[k',R_U]}_n|Q^{[k',R_U]}_n, W_k, \mathcal{R}_U')$$ (42)

$$\quad + H(\mathcal{R}_U')$$ (43)

$$\quad = H(A^{[k,R_U]}_1|\mathcal{F}, \mathcal{R}_U') - H(A^{[k',R_U]}_n|Q^{[k',R_U]}_n, \mathcal{R}_U')$$ (44)
By summing (45) over all $n \in [1 : N]$, we obtain the following relationship, which completes the proof,

$$N L \leq N H(A_{1:N}^{[k,R_U]} | F, R_U) - \sum_{n=1}^{N} H(A_{n}^{[k,R_U]} | F, R_U)$$

$$+ N H(R_U)$$

$$\leq (N - 1) H(A_{1:N}^{[k,R_U]} | F, R_U) + N H(R_U)$$

$$\leq (N - 1) \sum_{n=1}^{N} H(A_{n}^{[k,R_U]} | F, R_U) + N H(R_U)$$

$$\leq (N - 1) D + N H(R_U)$$

Proof: Starting with the database privacy constraint (10),

$$0 = I(W_k; F, A_{1:N}^{[k,R_U]}, R_U)$$

$$\geq I(W_k; A_{1:N}^{[k,R_U]}, R_U)$$

$$= I(W_k; A_{1:N}^{[k,R_U]}, R_U, F) + I(W_k; W_k | F, A_{1:N}^{[k,R_U]}, R_U)$$

$$= I(W_k; A_{1:N}^{[k,R_U]}, R_U, F)$$

$$= I(W_k; A_{1:N}^{[k,R_U]}, W_k, R_U) + I(W_k; W_k | F)$$

$$\geq I(W_k; A_{1:N}^{[k,R_U]} | F, W_k, R_U)$$

$$= I(W_k; A_{1:N}^{[k,R_U]} | F, W_k, R_U, F)$$

$$\geq I(W_k; A_{1:N}^{[k,R_U]} | F, W_k, R_U)$$

$$\geq I(W_k; A_{1:N}^{[k,R_U]} | F, R_U)$$

$$= I(W_k; A_{1:N}^{[k,R_U]} | F, R_U)$$

$$= H(A_{1:N}^{[k,R_U]} | F, R_U) - H(R_U | F, W_{1:k}, R_U)$$

$$\geq H(A_{1:N}^{[k,R_U]} | F, W_k, R_U) - H(R_U | F, W_k, R_U)$$

$$\geq H(A_{1:N}^{[k,R_U]} | F, W_k, R_U) - H(R_U | F, W_k, R_U)$$

$$\geq H(A_{1:N}^{[k,R_U]} | F, W_k, R_U) - H(R_U | F, W_k, R_U)$$

$$= H(A_{1:N}^{[k,R_U]} | F, W_k, R_U) - H(R_U | F, W_k, R_U)$$

$$= H(A_{1:N}^{[k,R_U]} | F, W_k, R_U) - H(R_U | F, W_k, R_U)$$

$$= H(A_{1:N}^{[k,R_U]} | F, W_k, R_U) - H(R_U | F, W_k, R_U)$$

$$= H(A_{1:N}^{[k,R_U]} | F, W_k, R_U) - H(R_U | F, W_k, R_U)$$

$$= H(A_{1:N}^{[k,R_U]} | F, W_k, R_U) - H(R_U | F, W_k, R_U)$$

$$= H(A_{1:N}^{[k,R_U]} | F, W_k, R_U) - H(R_U | F, W_k, R_U)$$

where (66) follows from the steps between (41)-(44) by applying Lemma 5 to Lemma 8 again.

By summing (66) over all $n \in [1 : N]$, we obtain the following relationship, which completes the proof,

$$0 \geq \sum_{n=1}^{N} H(A_{n}^{[k,R_U]} | Q_{n}^{[k,R_U]}, R_U) - N H(R_S)$$

$$\geq H(A_{1:N}^{[k,R_U]} | F, Q_{n}^{[k,R_U]}, R_U) - N H(R_S)$$

$$\geq \frac{N}{N-1} L - \frac{N}{N-1} H(R_U) - N H(R_S)$$

VI. Achievability Proof

Our achievability is based on the principle of converting a given PIR scheme into a valid SPIR scheme using the server-side and user-side common randomness in a manner that does not compromise the download cost; e.g., [52]. To that goal, the common randomness added to the desired symbols are substracted out as they are available at the user side, and the remaining common randomness unknown to the user protects the undesired messages. The challenge is to implement this for all possible user-side common randomness realizations which are unknown ahead of time. Steps of our achievable scheme:

1) Initial PIR query generation: For given $N$ and $K$, generate an initial PIR query table for each desired message using the scheme in [3], e.g., Tables I-II without $S_i$s.

2) Server-side common randomness assignment: For each desired message and each permutation of message index (e.g., 288 permutations in Example 2), mix all 1-sum symbols from the desired message across all the databases with the same new common randomness. We call it seed common randomness (e.g., $S_1$ in first three rows of Table II). Assign a new distinct common randomness to every 1-sum symbol from the undesired messages. For every $k$-sum symbol containing a desired message symbol, mix it with the common randomness from the $(k-1)$-sum symbol having the same $k-1$ undesired message symbols queried at another database. For every $k$-sum symbol not containing any desired message symbol, assign a new distinct common randomness. Repeat this until $k$ reaches $N$. Next, we permute non-seed common randomness indices (e.g., 4th-6th rows of Table II). We call this whole modified query table a query cell.

3) Server-side common randomness cycling: While keeping each query cell, create a new one by adding 1 (mod $|R_S|$) to each common randomness index (e.g., $S_1$ becomes $S_2$ in Table II). Repeat it $|R_S|$ times such that each query cell has a different seed common randomness index.

4) Query cell determination: The user has $|R_U|$ server-side common randomness. The user determines the query cell to be invoked, and selects a random permutation within that cell, by matching its user-side common randomness to the seed common randomness of the cell.

In this scheme, the message length $L$ is $N^K$ as in [3], the total amount of server-side common randomness required $|R_S|$ is $1 + \cdots + N^{K-1}$ and the total amount of user-side common randomness required $|R_U|$ is 1.