Noisy Private Information Retrieval

Karim Banawan  Sennur Ulukus
Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742
kbanawan@umd.edu ulukus@umd.edu

Abstract—We consider the problem of noisy private information retrieval (NPIR) from $N$ non-communicating databases, each storing the same set of $M$ messages. In this model, the answer strings are not returned through noiseless bit pipes, but rather through noisy memoryless channels. We aim at characterizing the PIR capacity for this model as a function of the statistical information measures of the noisy channels. We derive a general upper bound for the retrieval rate in the form of a max-min optimization. We use the achievable schemes for the PIR problem under asymmetric traffic constraints and the random coding arguments to derive a general lower bound for the retrieval rate. The upper and lower bounds match for $M = 2$ and $M = 3$, for any $N$, and any noisy channel. The results imply that separation between channel coding and retrieval is optimal except for adapting the traffic ratio from the databases.

I. INTRODUCTION

In the era of big data, efficient data-mining techniques are present everywhere, from social media to online-shopping and search history. These new challenges motivate studying the privacy issues that arise in modern networks. Private information retrieval (PIR), introduced by Chor et al. [1] and remained an important research avenue in computer science community (e.g., [2]–[4]), is a canonical problem to study the privacy of the downloaded content from public databases. In classical PIR, a user wishes to retrieve a file privately from $N$ distributed and non-colluding databases storing the same set of $M$ messages (files), in a way that no database can learn the identity of the user’s desired file. To that end, the user submits queries for the databases that do not reveal the user’s interest in the desired file. The databases respond with correct answer strings via noiseless links, from which the user reconstructs the desired file. PIR schemes are designed to be more efficient than the trivial scheme of downloading all the files stored in the databases in terms of the retrieval rate, which is the ratio between the number of downloaded bits from the desired message and the total download.

Recently, the PIR problem has attracted a renewed interest within the information theory community [5]–[9]. In the effort of characterizing the fundamental limits of the problem, Sun-Jafar introduced the notion of PIR capacity $C_{PIR}$ in [10], which is defined as the supremum of all PIR rates over all achievable retrieval schemes. [10] proved that for the classical PIR model, $C_{PIR} = (1 + \frac{1}{N} + \cdots + \frac{1}{N^{M-1}})^{-1}$. The achievability scheme is a greedy algorithm that employs a symmetric query structure for all databases. Following [10], the capacities of many interesting variants of the classical PIR problem have been considered, such as [11]–[36].

In all previous works, the links from the databases to the user are assumed to be noiseless. This assumption may not be valid in practice. For instance, while browsing (retrieving) the internet, some packets may be dropped randomly. This scenario can be abstracted by passing the answer strings through an erasure channel. Alternatively, the data packets may be randomly corrupted, which can be modeled as a binary symmetric channel that flips randomly some positions of the answer strings. Consequently, a more realistic setting is to assume that the databases return their answer strings through memoryless noisy channels with known transition probabilities. The noisy nature of the channel induces random errors along the received answer strings. This poses many interesting questions, such as: How to devise schemes that mitigate the errors introduced by the channel with a small sacrifice from the retrieval rate? Is there a separation between the channel coding needed for reliable transmission over noisy channels and the private retrieval scheme, or is there a necessity for joint processing? How do the statistical properties of the noisy channels fundamentally affect the retrieval rate?

In this paper, we introduce the noisy PIR (NPIR) problem. In noisy PIR, the $n$th database is connected to the user via a discrete memoryless channel with known transition probability distribution $p(y_n|x_n)$. Hence, the user needs to decode the desired message reliably by observing the noisy versions of the returned answer strings. Intuitively, since a channel with worse channel condition needs a lower code rate to combat the channel errors, we do not expect the lengths of the answer strings to be the same from all the databases. Therefore, in this work, we allow the traffic from each database to be asymmetric as in [36]. In this work, we aim at characterizing the capacity of the noisy PIR problem in terms of the statistical information measures of the noisy channels, the number of messages $M$, and the number of databases $N$. To that end, we first derive a general upper bound for the retrieval rate in the form of a max-min problem. The converse proof is inspired by the converse proof in [35], in particular in the way the asymmetry is handled. We show the achievability proof by random coding arguments and enforcing the uncoded responses to operate on one of the corner points of the PIR problem under asymmetric traffic constraints. The upper and lower bounds match for $M = 2$ and $M = 3$ messages, for arbitrary $N$ databases, and any noisy channel. Our results show that the channel coding needed to mitigate the channel errors and the retrieval scheme

This work was supported by NSF Grants CNS 13-14733, CCF 14-22111, CNS 15-26608 and CCF 17-13977.
are almost separable in the sense that the noisy channel affects only the traffic ratio requested from each database and not the explicit coding technique. Interestingly, the upper and lower bounds depend only on the capacity of the noisy channels and not on the explicit transition probability of the channels. We only provide sketches of the proofs here due to space limitations; proof details, illustrative remarks, extra examples and some figures can be found in the longer version [37].

II. SYSTEM MODEL

Consider a classical PIR model with $N$ replicated and non-communicating databases storing $M$ messages. Each database stores the same set of messages $W_{1:M} = \{W_1, \ldots, W_M\}$. The $n$th message $W_n$ is an $L$-length vector picked uniformly from $\mathbb{F}_2^L$. The messages $W_{1:M}$ are i.i.d., i.e., for $m \in \{1, \ldots, M\}$, 

$$H(W_n) = L, \quad H(W_{1:M}) = ML$$  

(1)

In PIR, in order to retrieve $W_i$, the user submits $N$ queries $Q_{i:1}^N = \{Q_i^1, \ldots, Q_i^N\}$, one for each database. The queries and the messages are statistically independent because the user has no knowledge about $W_{1:M}$.

$$I(W_{1:M}; Q_{i:1}^N) = 0, \quad i \in \{1, \ldots, M\}$$  

(2)

The $n$th database responds with a $t_n$-length answer string $A_n^i = (X_{n,1}^i, \ldots, X_{n,t_n}^i)$. The $n$th answer string is a deterministic function of $(W_{1:M}, Q_{i:1}^N)$, i.e., for all $n$ and $i$,

$$H(A_n^i|W_{1:M}, Q_{i:1}^N) = 0$$  

(3)

In noisy PIR, the user receives the $n$th answer string via a discrete memoryless channel with a transition probability $p(y_n|x_n)$. Thus, the user receives a noisy answer string $\bar{A}_n^i = (Y_{n,1}^i, \ldots, Y_{n,t_n}^i)$. Therefore, we have,

$$P(\bar{A}_n^i|A_n^i) = \prod_{n_1=1}^{t_n} p(y_{n,n_1}|x_{n,n_1})$$  

(4)

Denote the channel capacity of the $n$th response channel by $C_n$, where $C_n = \max_{p(x_n)} I(X_n; Y_n)$ and $X_n, Y_n$ are the single-letter input-output for the $n$th channel. Let the channel capacities be ordered such that $C_1 \geq C_2 \geq \cdots \geq C_N$ and $C = (C_1, \ldots, C_N)$ be the vector of the channel capacities.

The user and the databases agree on suitable lengths $\{t_n\}_{n=1}^N$ for the answer strings, which may be different in general. Define the traffic ratio vector $\tau = (\tau_1, \ldots, \tau_N)$, where $\tau_n = \frac{t_n}{\sum_{j=1}^N t_j}$, $n \in \{1, \ldots, N\}$.

To ensure the privacy, $Q_{i:1}^N$ should not reveal any information about $i$. We can write the privacy constraint as,

$$(Q_{i:1}^N, A_{1:M}) \sim (Q_{j:1}^N, A_{j:1}^N, W_{1:M}), \quad i, j \in \{1, \ldots, M\}$$  

(5)

In addition, the user should reconstruct the desired message $W_i$ by observing $\bar{A}_i^N$ with arbitrarily small probability of error $P_e(L)$, i.e., $P_e(L) \to 0$ as $L \to \infty$, hence

$$H(W_i|Q_{i:1}^N, A_{1:N}^i) \leq 1 + P_e(L) \cdot L = o(L)$$  

(6)

where $\frac{o(L)}{L} \to 0$ as $L \to \infty$.

The retrieval rate $R(\tau, C)$ is achievable if there exists a sequence of retrieval schemes, indexed by $L$, that satisfy (5), (6) with answer string lengths $\{t_n\}_{n=1}^N$ that yield $\tau$, thus,

$$R(\tau, C) = \lim_{L \to \infty} \frac{L}{\sum_{n=1}^N t_n}$$  

(7)

Consequently, the retrieval rate $R(\tau, C)$ is the supremum of $R(\tau, C)$ over all traffic ratio vectors in $T = \{(\tau_1, \ldots, \tau_N) : \tau_n \geq 0 \quad \forall n, \sum_{n=1}^N \tau_n = 1\}$. The PIR capacity $C_{PIR}(C)$ is sup $R(\tau, C)$ over all achievable retrieval schemes.

III. MAIN RESULTS AND DISCUSSIONS

Theorem 1 (Upper bound) For noisy PIR, $C_{PIR}(C)$ is upper bounded by $C(C)$ which is given by:

$$\max_{\tau \in \mathbb{T} \cap C} \min_{n \in [N]} \frac{\theta(0) + \frac{\theta(n_1)}{n_1} + \frac{\theta(n_2)}{n_2} + \cdots + \frac{\theta(n_M-1)}{n_M-1}}{1 + \frac{1}{n_1} + \frac{1}{n_2} + \cdots + \frac{1}{n_M-1}}$$  

(8)

where $\theta(\ell) = \sum_{n=\ell+1}^N \tau_n C_n$.

The proof of this upper bound is given in Section IV.

Theorem 2 (Lower bound) For noisy PIR, for a monotone non-decreasing sequence $n = (n_1, \ldots, n_M) \subset \{1, \ldots, N\}$, let $n_1 = 0$, and $S = \{i \geq 0 : n_i - n_{i-1} > 0\}$. Denote $y_{i,k}$ to be the number of stages of the achievable scheme that downloads $k$-sums from the $n$th database in one repetition of the scheme, such that $n_{k-1} \leq n \leq n_k$, and $\ell \in S$. Let $\xi_{\ell} = \sum_{i \in S \setminus \{\ell\}} (M-2)$. The number of stages $y_{i,k}$ is characterized by the following system of difference equations:

$$y_{0,k} = (n_0 - 1)y_{0,k-1} + \sum_{j \in S \setminus \{0\}} (n_j - n_{j-1})y_{j,k-1}$$  

$$y_{1,k} = (n_1 - n_0)y_{1,k-1} + \sum_{j \in S \setminus \{1\}} (n_j - n_{j-1})y_{j,k-1}$$  

$$y_{\ell,k} = n_0\xi_{\ell-1}y_{\ell-1,k-1} + \sum_{j \in S \setminus \{\ell\}} (n_{j-1} - n_{j-2})y_{j,k-1}$$  

(9)

where $\delta[\cdot]$ denotes the Kronecker delta function. The initial conditions of (9) are $y_{0,1} = \sum_{i \in S} (M-2)$, and $y_{i,k} = 0$ for $k \leq j$. Then, the achievable rate corresponding to $n$ is:

$$R(n, C) = \sum_{\ell \in S} \sum_{k=1}^{M-1} \frac{(M-1-k)y_{\ell,k}(n_{\ell} - n_{\ell-1})}{\sum_{i \in S} n_i}$$  

(10)

Consequently, the capacity $C_{PIR}(C)$ is lower bounded by:

$$C_{PIR}(C) \geq R(n, C) = \max_{n_0 \leq \cdots \leq n_{M-1} \in \{1, \ldots, N\}} R(n, C)$$  

(11)

The proof of Theorem 2 is given in Section V. We note that the bounds on $C_{PIR}(C)$ do not depend explicitly on the transition probability of the channels $p(y_n|x_n)$, but rather depend on the capacity of the noisy channels $C_n$. Furthermore, the bounds imply that the channel coding needed for combating channel errors is almost separable from the retrieval scheme.
The channel coding problem and the retrieval problem are coupled only through agreeing on a traffic ratio vector \( \tau \). Other than \( \tau \), the channel coding acts as an outer code for the responses of the databases to the user queries.

**Corollary 1 (Capacity for \( M = 2, 3 \) messages)** For noisy PIR, the capacity \( C_{PIR}(C) \) for \( M = 2 \), and a fixed \( N \) is:

\[
C(C) = \max_{n_i \in [N]} \sum_{n=1}^{N} \frac{n_0 n_1}{C_n} + \sum_{n=n_0+1}^{n_1} \frac{n_0 n_1}{C_n}
\]

and for \( M = 3 \), \( C(C) \) is given by,

\[
\max_{n_i \in [N]} \sum_{n=1}^{n_0} \frac{n_0 n_1 n_2}{C_n} + \sum_{n=n_0+1}^{n_1} \frac{n_0 n_1 n_2}{C_n} + \sum_{n=n_1+1}^{n_2} \frac{n_0 n_1 n_2}{C_n}
\]

The proof of Corollary 1 follows from the optimality of PIR-WTC-II scheme in [36] by replacing \( 1 - \mu_n \) by \( C_n \).

**IV. Converse Proof**

The main idea of the converse hinges on the fact that the traffic from the databases should be dependent on the relative channel qualities (i.e., channel capacities) of the response channels. Thus, we extend the converse proof in [35] to account for the noisy observations. We will need the following lemmas, whose proofs can be found in [37].

**Lemma 1 (Interference bound)** For NPIR, the mutual information between the interfering messages \( W_{2:M} \) and the noisy answers \( \tilde{A}^{[1]}_{1:N} \) given the desired message \( W_1 \) is bounded by,

\[
I\left(W_{2:M}; \tilde{Q}^{[1]}_{1:N}, \tilde{A}^{[1]}_{1:N} | W_1 \right) \leq \sum_{n=1}^{N} t_n C_n - L + o(L)
\]

**Lemma 2 (Induction lemma)** For all \( m \in \{2, \cdots, M\} \) and for an arbitrary \( n_{m-1} \in \{1, \cdots, N\} \), the mutual information term in Lemma 1 can be inductively lower bounded as,

\[
I\left(W_{m:M}; \tilde{Q}^{[m]}_{1:N}, \tilde{A}^{[m]}_{1:N} | W_{1:m-1} \right) \leq \frac{1}{n_{m-1}} I\left(W_{m+1:M}; \tilde{Q}^{[m]}_{1:N}, \tilde{A}^{[m]}_{1:N} | W_{1:m-1} \right) + \frac{1}{n_{m-1}} \left( L - \sum_{n=n_{m-1}+1}^{N} t_n C_n \right) - o(L)
\]

Now, we derive an explicit upper bound for the retrieval rate. Fixing the length of the \( n \)th answer string to \( t_n \), and applying Lemma 1 and Lemma 2 successively for an arbitrary sequence \( \{n_i\}_{i=1}^{M-1} \subset \{1, \cdots, N\}^{M-1} \), we have the following,

\[
\sum_{n=1}^{N} t_n C_n - L + \tilde{o}(L)
\]

\[
\geq \frac{1}{n_1} \left( L - \sum_{n=n_1+1}^{N} t_n C_n \right) + \frac{1}{n_1} I\left(W_{3:M}; \tilde{Q}^{[2]}_{1:N}, \tilde{A}^{[2]}_{1:N} | W_{1:2} \right)
\]

**V. Achievability Proof**

We begin first by a motivating example of PIR from BSC(0.1) and BSC(0.2) for \( M = 3 \). For this example, we provide an explicit scheme that uses achievability of Shannon’s channel coding theorem for BSC using linear block codes [38]. We extend the scheme for arbitrary \( M \), \( N \), and any noisy response channels, by means of the random coding argument.

**A. Example: \( M = 3, N = 2 \), via BSC(p_1), BSC(p_2)**

1) **Explicit Upper Bound**: From Theorem 1, (8) can be explicitly written as the following linear program:

\[
\max_{\tau_1, \tau_2, R} \frac{R}{b_2 - a_2}
\]

s.t. \( R \leq \frac{1}{3} \left( \frac{1-H(p_1)}{3} - \frac{1-H(p_2)}{3} \right) \tau_1 \)

\( R \leq \frac{5}{9} \left( \frac{1-H(p_1)}{5} - \frac{1-H(p_2)}{5} \right) \tau_2 \)

\( R \leq \frac{4}{7} \left( \frac{1-H(p_1)}{4} - \frac{1-H(p_2)}{7} \right) \tau_2 \)

\( 0 \leq \tau_2 \leq 1 \)

(21)

The optimal solution of (21) resides at one of the corner points of the feasible set, hence an upper bound for \( C_{PIR}(p_1, p_2) \) is:

\[
\max \left\{ \frac{1-H(p_1)}{3}, \frac{2}{1-H(p_1)} + \frac{1}{1-H(p_2)}, \frac{4}{1-H(p_1)} + \frac{4}{1-H(p_2)} \right\}
\]

(22)
2) Achievable Scheme for BSC(0.1), BSC(0.2): The explicit upper bound in (22) implies that \( R \leq \frac{4}{1-H(P)} \), which is 0.218 for \( p_1 = 0.1, p_2 = 0.2 \). To achieve this rate, we enforce the ratio between the uncoded traffic, i.e., before channel coding, to be 4 : 3. This results in the explicit code rate of \( \frac{4}{1-H(P)} - \frac{3}{1-H(P)} \), which appears in the denominator of the upper bound. Concurrently, this results in retrieving 4 desired bits per scheme repetition, which appear at the numerator.

The user repeats the following retrieval scheme for \( \nu \) times. Each repetition of the scheme operates over \( L^* = 4 \) bits from \( W_{1,3} \). The user permutes the indices of the bits of each message independently and uniformly. Let \( a_i(j), b_i(j), c_i(j) \) denote the \( i \)th bit of block \( j \) from the permuted message \( W_1, W_2, W_3 \), respectively. Assume that the desired file is \( W_1 \). In block \( j \), the user requests to download \( a_1(j), b_1(j), \) and \( c_1(j) \) from database 1. From database 2, the user exploits the side information generated from database 1 by requesting the sums \( a_2(j) + b_1(j), a_3(j) + c_1(j), \) and \( b_2(j) + c_2(j) \). Finally, the user exploits the side information generated from database 2 by downloading \( a_4(j) + b_2(j) + c_2(j) \) from database 1. The query table for the \( j \)th block is summarized in Table I. Denote the number of uncoded bits requested from the \( n \)th database by \( D_n \), then \( D_1 = 4, D_2 = 3 \). This guarantees that the ratio between the uncoded traffic is 4 : 3. This query structure is private, as all combinations of the sums are included in the queries and the message bits are randomly permuted.

<table>
<thead>
<tr>
<th>Database 1</th>
<th>Database 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_1(j) )</td>
<td>( a_2(j) + b_1(j) )</td>
</tr>
<tr>
<td>( b_1(j) )</td>
<td>( a_3(j) + c_1(j) )</td>
</tr>
<tr>
<td>( c_1(j) )</td>
<td>( b_2(j) + c_2(j) )</td>
</tr>
<tr>
<td>( a_4(j) + b_2(j) + c_2(j) )</td>
<td></td>
</tr>
</tbody>
</table>

The \( n \)th database concatenates the uncoded responses to the user’s queries into a vector \( U_{n}^{[1]} \) of length \( \nu D_n \), i.e.,
\[
U_{1}^{[1]} = [a_1(1), b_1(1), c_1(1), a_4(1) + b_2(1) + c_2(1), \ldots, a_1(\nu), b_1(\nu), c_1(\nu), a_4(\nu) + b_2(\nu) + c_2(\nu)]^T
\]
\[
U_{2}^{[1]} = [a_2(1) + b_1(1), a_3(1) + c_1(1), b_2(1) + c_2(1), \ldots, a_2(\nu) + b_1(\nu), a_3(\nu) + c_1(\nu), b_2(\nu) + c_2(\nu)]^T
\]

Using Shannon’s theorem for BSC(\( p \)) [38, Theorem 4.1.7], for all \( p \in (0,1) \), all but less than \( \rho \) linear \([n,k]\) codes, \( C \), satisfy \( P_e(C) < 2^{-2^{-n}(1-pR)} \), for some \( \Delta(p, R) > 0 \), if \( R = \frac{k}{n} < 1 - H(p) \). The \( n \)th database picks a \((t_n, \nu D_n)\) linear block code, which belongs to the set of these good codes. The \( n \)th database encodes the vector \( U_{n}^{[1]} \) to a coded answer string \( A_{n}^{[1]} \) of length \( t_n \) such that \( t_n = \frac{\nu D_n}{1-H(P)} \). This ensures that \( \frac{\nu D_n}{t_n} < 1 - H(p_n) \). The \( n \)th database responds with \( A_{n}^{[1]} \) via the noisy channel BSC(\( p_n \)). The user receives the noisy answer string \( \hat{A}_{n}^{[1]} \) from the \( n \)th database.

For the decoding, the user employs the nearest-codeword decoder to find an estimate of \( A_{n}^{[1]} \) based on \( \hat{A}_{n}^{[1]} \). Using union bound, the probability of error in decoding is bounded by:
\[
P_e(L) \leq \sum_{n=1}^{2} P_e(C_n) \leq \sum_{n=1}^{2} 2^{-t_n \Delta(p_n, \frac{\nu D_n}{t_n})}
\]

As \( \nu \to \infty, L \to \infty \) and \( t_n \to \infty \), hence \( P_e(L) \to 0 \). This ensures the decodability of \( U_{n}^{[1]} \) with high probability. Since the vectors \( U_{1}^{[1]}, U_{2}^{[1]} \) are designed to exploit the side information, the user can cancel the undesired messages and be left only with the correct \( W_1 \) with probability of error \( P_e(L) \).

The retrieval scheme decodes \( L = \nu L^* = 4\nu \) bits from the desired messages. The retrieval scheme downloads \( t_n = \frac{\nu D_n}{1-H(P)_{n}} \) from the \( n \)th database, hence as \( \nu \to \infty \), we have
\[
R = \frac{L}{t_1 + t_2} = \frac{\nu D_1}{1-H(P_{1})} + \frac{\nu D_2}{1-H(P_{2})} = \frac{4}{1-H(P_{2})} + \frac{3}{1-H(P_{2})}
\]

B. General Achievable Scheme

The main idea of the general scheme is to use the uncoded response from the \( n \)th database to user’s query as an index for choosing the transmitted codeword from a codebook generated according to the optimal probability distribution. The uncoded query structure is derived from one of the achievable schemes for the corner points of the asymmetric PIR problem [35].

As in [35], we denote the number of side information symbols that are used simultaneously in the initial round at the \( n \)th database by \( s_n \in \{0, 1, \ldots, M - 1\} \). For a given non-decreasing sequence \( \{a_n\}_{M-1} \subset \{1, \ldots, N\}^M \), the databases are divided into groups, such that group \( \ell \) includes databases \( n_{\ell-1} + 1 : n_{\ell} \). Let \( s_n = i \) for all databases belonging to the \( i \)th group. Denote \( S = \{i : s_n = i \} \) for some \( n \in \{1, \ldots, N\} \). Similar to [20], the \( k \)th round is the download queries that admit a sum of \( k \) different messages (\( k \)-sum [10]). A stage of the \( k \)th round is a query block that exhausts all \( \binom{M}{k} \) combinations of the \( k \)-sum. Denote \( y_k[k] \) to be the number of stages in round \( k \) downloaded from the \( n \)th database, such that \( n_{\ell-1} + 1 \leq n \leq n_{\ell} \). Our scheme is repeated for \( \nu \) repetitions, each having the same query structure and operating over message symbols of length \( L^* \). Denote the total requested symbols from the \( n \)th database in one repetition of the scheme by \( D_n(n) \). The details of the achievable scheme are:

1) Codebook construction: According to the optimal probability distribution \( p^*(x_n) \) (that maximizes \( I(X_n; Y_n) \)), the \( n \)th database constructs a \((2^\nu D_n(n), t_n(n))\) codebook \( C_n \), where \( t_n(n) = \frac{\nu D_n(n)}{C_n} \), at random, i.e.,
\[
p(x_{n,1}, \ldots, x_{n, t_n(n)}) = \prod_{j=1}^{t_n(n)} p^*(x_{n,j}).
\]
Specifically, the codebook \( C_n \) can be written as:
\[
\begin{bmatrix}
    x_{1}(1) & x_{2}(1) & \cdots & x_{t_n(n)}(1) \\
    x_{1}(2) & x_{2}(2) & \cdots & x_{t_n(n)}(2) \\
    \vdots & \vdots & \ddots & \vdots \\
    x_{1}(2^\nu D_n(n)) & \cdots & x_{t_n(n)}(2^\nu D_n(n))
\end{bmatrix}
\]

2) Initialization at the user side: The user permutes each message independently and uniformly.
3) Initial download: From the $n$th database, $1 \leq n \leq n_0$, the user requests $\prod_{i \in S} \left( \frac{M-2}{s_i-1} \right)$ symbols from the desired message and sets the round index $k = 1$.

4) Message symmetry: For each stage initiated in the previous step, the user completes the stage by requesting the $\left( \frac{k-1}{s_i-1} \right)$ undesired $k$-sum combinations.

5) Database symmetry: The user repeats step 4 over each group of databases such that the scheme is symmetric within the same group of databases.

6) Exploitation of side information: The undesired symbols downloaded within the $k$th round are used as side information in the $(k+1)$th round. The user requests $(k+1)$-sum consisting of 1 desired symbol and a $k$-sum of undesired symbols only that were generated in the $k$th round. The $n$th database belonging to the $n$th group exploits the side information generated in the $k$th round from all databases except itself if $s_n \leq k$. Moreover, for $s_n = k$, extra side information can be used by constructing $k$-sums of the undesired symbols in round 1 from the databases in group 0.

7) Repeat steps 4, 5, 6 after setting $k = k + 1$ until $k = M$.

8) Repeat steps 3, · · · , 7 for a total of $\nu$ repetitions.

9) Shuffle the order of the queries uniformly.

10) Encoding the responses to the user queries: The $n$th database concatenates all the responses to the user queries in a vector $U_n^{[i]}$ of length $\nu D_n(n)$. The $n$th database uses $U_n^{[i]}$ as an index for choosing codeword from $C_n$, i.e., the index of the codeword and $U_n^{[i]}$ should be in bijection (e.g., by transforming $U_n^{[i]}$ into a decimal value). Consequently, the $n$th database responds with,

$$A_n^{[i]} = [x_1(U_n^{[i]}), x_1(U_n^{[i]}), \ldots, x_{\nu}(U_n^{[i]}))^T (28)$$

REFERENCES


