Protecting Data Privacy in Private Information Retrieval Schemes

Yuval Ishai * Eyal Kushilevitz[†]

Department of Computer Science
Technion, Haifa 32000, Israel
April 27, 1997

Abstract

Private Information Retrieval (PIR) schemes allow a user to retrieve the i-th bit of a data string x, replicated in $k \geq 2$ databases, while keeping the value of i private. The main cost measure for such a scheme is its communication complexity.

We study PIR schemes where in addition to the user's privacy we require *data privacy*. That is, in every invocation of the retrieval protocol the user learns exactly a single (physical) bit of x and no other information.

We present general transformations that allow translating PIR schemes satisfying certain properties into PIR schemes that respect data privacy as well, with a small penalty in the communication complexity. Using our machinery we are able to translate currently known PIR solutions into schemes satisfying the newly introduced, stronger privacy constraint. In particular we get: a k-database scheme of complexity $O(\log n \cdot n^{1/(2k-1)})$ for every $k \geq 2$; an $O(\log n)$-database scheme of poly-logarithmic complexity; and a 2-database computational PIR scheme of complexity $O(n^c)$, for every constant $c > 0$. All these schemes require only a single round of interaction.

*e-mail: yuvali@cs.technion.ac.il
[†]e-mail: eyalk@cs.technion.ac.il URL: http://www.cs.technion.ac.il/~eyalk

1 Introduction

Private Information Retrieval (PIR) schemes allow a user to retrieve information from a database in a way that maintains the identity of the data retrieved by the user hidden (in the information theoretic sense) from the database administrator. Formally, the database is viewed as an n-bit string x out of which the user wishes to retrieve the bit x_i while maintaining i private. The notion of PIR was introduces in [10], where it was shown that if there is only one copy of the database available then $\Omega(n)$ bits of communication are needed for (information theoretic) PIR. However, if there are $k \geq 2$ (separated) copies of the database available then there are solutions with much better communication complexity: $O(n^{1/3})$ bits for $k = 2$ databases; $O(n^{1/k})$ bits for $k \geq 3$ databases; and $O(\log^2 n \log\log n)$ bits for $k = O(\log n)$ databases. [1] showed an upper bound of $O(n^{1/(2k-1)})$ bits for $k \geq 3$ databases. The *computational* counterpart of the definition (i.e., PIR schemes where the privacy is only with respect to polynomial-time databases, relying on certain intractability assumptions) were considered in [9, 15]; they show how to obtain schemes with communication complexity $O(n^c)$ (for any constant $c > 0$) for $k = 2$ databases [9] (see also [16]) and even for the case of a single database [15] (based on a stronger assumption).

One disturbing property of the currently existing PIR schemes is that since the databases do not know what bit x_i the user wishes to retrieve, they provide the user with a lot of information out of which (by combining the answers from different databases) the user can compute x_i. However, in all these solutions the user is able to reconstruct other physical bits of the database (i.e., x_j for $j \neq i$) or other information about the database (e.g., the exclusive-or of certain subsets of the bits of x). For instance, in a single invocation of the 2-database scheme from [10] (which is the most efficient of its kind known to date), a user can systematically retrieve $O(n^{1/3})$ *physical* bits of data (see Section 4, Example 1). This means that the user gets much more information than what it is supposed to get, which is particularly undesirable in case of commercial databases where the user is supposed to pay by the amount of data it retrieves.

Therefore, we are interested in *symmetrically-private information retrieval* (or SPIR for short) schemes; i.e., schemes which respect the privacy of the *data* as well as the user's. That is, in addition to maintaining the user's privacy, any single invocation of such a stronger scheme does not allow the user (even a dishonest one) to obtain any information other than a single *physical* bit of the data. In the usual PIR setting the databases do not interact with each other; the only interaction allowed is between the user and the databases. It is easy to observe (see Appendix A) that in this basic setting, such stronger schemes cannot be realized at all, regardless of their complexity. We thus use a minimal extension of the basic setting: we still disallow direct interaction between the databases, but we grant them access to a *shared* random string, known to all databases but unknown to the user. This kind of extension has been studied before in the context of private computation [11] (see also [12]) and other scenarios such as non-interactive zero-knowledge [5]. Adding a resource of shared randomness to PIR schemes is particularly natural, since even in the basic PIR setting databases are required to maintain *identical* copies of the same data string.[1] Moreover, if one is willing to settle for *computational* privacy of the data (while maintaining the information-theoretic privacy of the user), sharing a short random seed allows generating longer shared pseudo-random strings without extra communication.

We prove that currently known PIR schemes (information theoretic and computational) can be transformed into SPIR schemes, with the same number of databases and at most a logarithmic factor of communication overhead. Our schemes maintain the general paradigm of existing PIR schemes: all databases hold an *identical* copy of x, and all protocols use a *single* queries-answers round. In particular, schemes from [10, 1, 9] translate into:

- k-database SPIR scheme of complexity $O(\log n \cdot n^{1/(2k-1)})$.
- $O(\log n)$-database SPIR scheme of complexity $O(\log^2 n \cdot (\log\log n)^2)$.
- 2-database computational SPIR of complexity $O(n^c)$, for every constant $c > 0$.

[1] In fact, we may view the data string as consisting of a deterministic part, containing ordinary data, and a random part containing the shared random string.

In case of an *honest but curious* user (i.e., a user which follows the prescribed scheme, but may try to deduce extra information from the communication), the logarithmic overhead can be avoided.

In a nutshell, our solutions work as follows. Consider any 1-round PIR scheme. The user, depending on the index i, produces k queries $q_1, \ldots, q_k$ (one per each database); it sends each of them to the corresponding database and in response receives k answer strings $a_1, \ldots, a_k$. Then, the user applies a *reconstruction function* f to obtain the desired bit x_i. Our idea is that it might be possible for the user to compute the value of f without actually getting the answers $a_1, \ldots, a_k$ but rather some other messages $m_1, \ldots, m_k$ that keep the privacy of the string x. This is a very similar scenario to the scenario of [11].[2] The problem here is that not all functions are known to have an efficient solution in the model of [11], and even if the reconstruction function f does have an efficient solution, in order to maintain the communication complexity of the PIR scheme that we start with we need a solution for f that will be linear (or "almost" linear). We present methods which overcome these difficulties in various ways, that allow dealing with the reconstruction functions of the existing PIR schemes and that seem general enough to apply for future PIR solutions as well.

Organization: In Section 2 we introduce notations and basic definitions. Section 3 includes SPIR schemes which rely on the user being honest. In Section 4 we present schemes which keep the data private from *any*, possibly dishonest, user (with a minor extra communication cost). Section 5 contains a discussion of our results and some of their possible generalizations and applications. Appendix A contains a proof of the impossibility of SPIR in the usual PIR setting (without interaction between the databases or shared randomness). Appendix B contains proofs which were deferred from the main body of text.

2 Definitions

The following notations and conventions are used throughout. $[\ell]$ denotes the set $\{1, 2, \ldots, \ell\}$. For any sets $S, S' \subseteq [\ell]$, we let $S \oplus S'$ denote the symmetric difference between S and S' (i.e., $S \oplus S' = (S \setminus S') \cup (S' \setminus S)$), and χ_S denote the *characteristic vector* of S, an ℓ-bit binary string whose j-th bit is equal to 1 iff $j \in S$. To simplify notation, $S \oplus j$ and χ_j are used instead of $S \oplus \{j\}$ and $\chi_{\{j\}}$, respectively. For any $\sigma \in \{0,1\}^d$, weight(σ) denotes the number of nonzero entries in σ. By default, whenever referring to a *random* choice of an element from a finite domain A, the associated distribution is uniform over A, and is independent of all other random choices.

The following notations and definitions are specifically related to PIR and SPIR schemes. We let k denote the number of databases, and $\mathcal{DB}_j$ denote the j-th database. x denotes an n-bit data string which is held by all k databases. We let i denote the position (also called *index*) of a data bit which the user wants to retrieve.

A *PIR* scheme is a randomized protocol, where in each round *queries* are sent from the user to each database, and *answers* are sent from each database to the user. At the end of the execution, the user applies some *reconstruction* function to the queries, answers and the index i, to obtain the desired data bit x_i. Every (1-private) PIR scheme must satisfy the following *user privacy* requirement: under any two indices i, i', the communication seen by any *single* database is identically distributed. (This information-theoretic privacy requirement is relaxed to computational indistinguishability in the case of *computational PIR*).

A *SPIR* scheme is a PIR scheme (in a setting which allows databases to access a *shared* random string not known to the user), such that in a single invocation the user cannot obtain any information which doesn't follow from some physical bit of data. Formally, this *data privacy* requirement implies that for *any* behavior of a user (possibly a dishonest one, not following the scheme's specification) there exists an index i, such that the distribution of communication between the user and the (honest) databases depends on x_i alone, i.e. is the same for every x, x' such that $x_i = x'_i$. An *honest-user-SPIR* scheme is a PIR scheme

[2] In the setting of [11] there are several players $P_1, \ldots, P_k$ where each P_i has some input y_i and they all share a random string. Each of them is supposed to send a single message m_i, based on its input and the shared random string, to a referee (named Carol). From the messages she received, Carol should be able to compute $f(y_1, \ldots, y_k)$ but should not be able to learn any additional information about the inputs.

that satisfies the data-privacy requirement with respect to an *honest* user, which follows the scheme's specification.

By default, the terms "PIR scheme" and "SPIR scheme" refer to *1-round*, information-theoretically private schemes (all of our SPIR schemes will require only a single round of interaction). Complexity is measured, by default, in terms of communication. The communication complexity of a k-database scheme will be denoted $(\alpha_k(n), \beta_k(n))$, where $\alpha_k(n)$ is the total number of query bits and $\beta_k(n)$ is the total number of answer bits on a data string of size n.

Finally, we use $\mathcal{B}_2$ to denote the 2-database covering-codes scheme from [10], and $\mathcal{B}_k$ to denote its k-database generalization from [1]. These schemes, which are the most efficient of their kind known to date, are described in subsection 3.3.

3 SPIR Schemes with Respect to Honest Users

In this section we construct SPIR schemes, which maintain the privacy of the user, as well as the privacy of the data against *honest but curious* users. The schemes we obtain in this case are more efficient than those obtained for the case of a dishonest user — all of them are of the same asymptotic communication complexity as the PIR schemes they are based on. Later, we will use similar techniques to deal with dishonest users.

3.1 A Simple Example

We start with a simple example, whose ideas will be subsequently generalized to handle more involved cases. Consider the elementary d-dimensional cube scheme from [10]. This is a PIR scheme (without data privacy) for $k = 2^d$ databases, which is described in the following.

Assume w.l.o.g. that the database size is $n = \ell^d$, where ℓ is an integer. The index set $[n]$ can then be identified with the d-dimensional cube $[\ell]^d$, in which each index $i \in [n]$ can be naturally identified with a d-tuple $(i_1, \ldots, i_d)$. A d-dimensional *subcube* is a subset $S_1 \times \cdots \times S_d$ of the d-dimensional cube, where each S_i is a subset of $[\ell]$. Such a subcube is represented by the d-tuple $C = (S_1, \ldots, S_d)$. The $k(= 2^d)$ databases will be indexed by all binary strings of length d. The scheme proceeds as follows.

Queries: The user picks a random subcube $C = (S_1^0, \ldots, S_d^0)$, where $S_1^0, \ldots, S_d^0$ are independent, random subsets of $[\ell]$. Let $S_m^1 = S_m^0 \oplus i_m$ $(1 \leq m \leq d)$. For each $\sigma = \sigma_1\sigma_2\cdots\sigma_d \in \{0,1\}^d$, the user sends to database $\mathcal{DB}_\sigma$ the subcube $C_\sigma = (S_1^{\sigma_1}, \ldots, S_d^{\sigma_d})$, where each set $S_m^{\sigma_m}$ is represented by its characteristic ℓ-bit string.

Answers: Each database $\mathcal{DB}_\sigma$, $\sigma \in \{0,1\}^d$, exclusive-ors the data bits residing in the subcube C_σ, and sends the resultant bit b_σ to the user.

Reconstruction: The user computes x_i as the exclusive-or of the k bits b_σ it has received.

The scheme's correctness follows from the fact that every bit in x except x_i appears in an even number of subcubes C_σ, $\sigma \in \{0,1\}^d$, and x_i appears in exactly one such subcube (see [10] for details).

Since the scheme doesn't use shared randomness, Proposition 3 (see Appendix A) implies that it must expose some extra information, which doesn't follow from x_i, to the honest user. Indeed, the user receives the exclusive-or of k different subsets of data bits. In this case, the extra information can be eliminated by applying the following simple modification to the above scheme. Instead of sending the original answer b_σ, each database $\mathcal{DB}_\sigma$ will send a *masked* answer $b_\sigma \oplus r_\sigma$, where $r = r_{0\cdots00}r_{0\cdots01}\cdots r_{1\cdots10}$ is a $(k-1)$-bit *shared* random string, and $r_{1\cdots11}$ is computed as the exclusive-or of the bits of r. (Alternatively, $r_{0\cdots00}, r_{0\cdots01}, \ldots, r_{1\cdots11}$ can be randomly chosen from the k-tuples whose bits xor to 0). Under the modified scheme, an honest user's view is uniformly distributed among all k-tuples that xor to $\bigoplus_{\sigma\in\{0,1\}^d} b_\sigma$, which by the scheme's correctness equals the physical bit x_i.

We note that even in this simple case, a dishonest user may obtain information that doesn't follow from any physical bit of data. Specifically, by choosing the cubes C_σ not according to the scheme's specification, a user may obtain the exclusive-or of different (possibly large) subsets of data bits. Such a dishonest user is treated in Section 4.

3.2 SPIR Schemes Based on PSM Protocols

The specific reconstruction function of the previous scheme (namely, a Boolean xor of k bits) makes it easy for the databases to send randomized answers which disclose only the *output* of this function. Now, what can be said about the difficulty of this problem in general? Precisely this problem is captured by the model of private computation introduced by [11] and further studied in [12]. We call this model the *PSM ("Private Simultaneous Messages")* model. In this model there are k players, each player P_j holding a private input string y_j, and an external referee called Carol. All players have access to a shared random input, which is unknown to Carol. Based on its private input y_j and the shared random input, each player P_j simultaneously sends a single message to Carol. From the messages she received, Carol should be able to compute some predetermined function $f(y_1, \ldots, y_k)$ of the inputs, but should obtain no additional information on the input string other than what follows from the value of f. The *PSM complexity* of f is the number of communication bits needed to privately compute the function f in such a way. We denote this complexity by $c_k(n)$, where n is the total number of input bits held by the k players. In [11, 12] several upper bounds on PSM complexity are obtained. These include an $O(k \cdot S(n)^2)$ bound for any function with a deterministic branching program of size $S(n)$ (for any partition of the n bits among the players), which in particular implies the same upper bound for every function with a Boolean formula of size $S(n)$.

Note that in order to use a PSM protocol for evaluating the reconstruction function, this function must depend only on the answers computed by the databases. Although this is the case in the elementary cube scheme described above, in the $\mathcal{B}_2$ scheme, for instance, the reconstruction function heavily depends on the index i held by the user. However, every PIR scheme may be efficiently transformed into a scheme which satisfies this requirement, e.g. by letting the user share the index i between two databases (so that its privacy is not violated), and concatenating these shares along with the queries to the original answers of the two databases.

Proposition 1. *Suppose $\mathcal{P}$ is a 1-round k-database PIR scheme with communication complexity $(\alpha_k(n), \beta_k(n))$, such that the reconstruction function f: (1) depends only on the answers sent by the databases, and (2) has PSM complexity of $c_k(n)$. Then there exists a 1-round SPIR scheme $\mathcal{S}$,* with respect to an honest user*, whose communication complexity is $(\alpha_k(n), c_k(\beta_k(n)))$.*

Proof. A scheme $\mathcal{S}$ of the specified complexity can be obtained from $\mathcal{P}$ as follows. The databases compute their answers as in $\mathcal{P}$, but instead of sending back their answers, they use their shared randomness to simulate the PSM computation of the reconstruction function, f. The correctness and privacy of $\mathcal{S}$ follow from the correctness and privacy of $\mathcal{P}$ and of the PSM protocol for f. $\square$

We note that if the PSM complexity of the reconstruction function is high, then some of the extra cost may be absorbed by originating from a PIR scheme whose communication is unbalanced (so that the databases send less bits than the user). To this end, balancing techniques from [10, 9] may be used.

Luckily, the reconstruction functions in some currently known PIR schemes (at least in their elementary unbalanced form) have the lowest PSM complexity possible, thus allowing their transformation into honest-user-SPIR schemes with the same communication complexity.

Theorem 1. *There exist:*

- $\lceil \log_2 n + 1 \rceil$*-database honest-user-SPIR scheme of communication complexity* $O(\log^2 n \log\log n)$*;*
- *2-database honest-user* computational *SPIR scheme of communication complexity $O(n^c)$ (for any $c > 0$, assuming the existence of a one-way function).*

Proof. The first SPIR scheme is obtainable from the elementary polynomial interpolation scheme of [10], and the second from the computational PIR scheme of [9]. We observe that in the first scheme, reconstruction consists of computing the free coefficient of a degree-$\lceil \log_2 n \rceil$ univariate polynomial (over a finite field) given its values at $\lceil \log_2 n \rceil + 1$ fixed, distinct points. In the second scheme, reconstruction consists of simply xoring 4 bits. Both of these reconstruction functions are *linear*, thus admitting PSM protocols of complexity $c_k(n) = n$ (a protocol for any linear function can be obtained by a straightforward generalization of the protocol for exclusive-or from subsection 3.1). Hence, by Proposition 1 we can obtain honest-user-SPIR schemes of the same complexity as the original PIR schemes, i.e. of the specified complexity. □

Since even some of the simplest Boolean functions (including functions with linear formula size) are not known to have sub-quadratic PSM complexity, using a PSM protocol for computing non-linear reconstruction functions may result in a considerable communication overhead. For instance, the $\mathcal{B}_2$ scheme from [10] seems to pose such a problem. Although in this case the problem can be tackled by applying Proposition 1 alone,[3] in the next subsection we introduce a different tool which gives conceptually simpler and more efficient (by a constant factor) schemes, and which will be more extensively used in the Section 4.

3.3 SPIR Schemes Bases on Conditional Disclosure of Secrets

The reconstruction functions in several important PIR schemes, including the $\mathcal{B}_2$ scheme and schemes obtained by applying communication balancing techniques, have a very special structure. These functions depend only on some (small) subset of the message bits, which is completely determined by the index i of the data bit being retrieved. When such a scheme is modified to allow the reconstruction to depend on the answers alone, some of the input to the modified reconstruction function will typically be known to the user. This motivates a generalization of the usual PSM setting, in which some of the inputs held by the players are also known to Carol, and some input bits may be known to several players. The original correctness requirement is relaxed so that Carol is allowed to compute the value of f from the messages she has received *and* the input she holds. The original privacy requirement is relaxed so that Carol is not allowed to obtain any information other than what follows from the value of f *and the input she holds.*

We now consider a special case of this general problem, which we call *conditional disclosure of secrets.* The problem of conditionally disclosing a secret s (subject to a condition f) is described in the following. Let $f : \{0,1\}^n \to \{0,1\}$ be some Boolean function (the "condition"). An n-bit input y is partitioned between k players, and Carol holds the *entire* string y as input. In addition, a secret input s (single bit unless otherwise mentioned) is known to at least one of the players, but is unknown to Carol. As in the general PSM setting, the players use their inputs and the shared randomness to send simultaneous messages to Carol, such that: (1) if $f(y) = 1$, then from her input y and the messages she received, Carol should be able to reconstruct the secret s; and (2) if $f(y) = 0$, then carol obtains no information (in the information-theoretic sense) about s.

The next proposition shows that the problem of disclosing a secret subject to the condition f reduces to the problem of generalized secret sharing [4, 13] relative to a corresponding access structure.[4]

Proposition 2. *Let $f_M : \{0,1\}^m \to \{0,1\}$ be a monotone Boolean function (access structure), and let $f : \{0,1\}^n \to \{0,1\}$ be a* projection *of f_M [18]. That is, $f(y_1, \ldots, y_n) = f_M(g_1, \ldots, g_m)$, where each g_i is a function of a single variable y_j. Suppose there exists a generalized secret sharing scheme $\mathcal{S}$ realizing the access structure f_M, in which the total size of the shares is c. Then there exists a protocol $\mathcal{P}$ for disclosing*

[3] Modify the $\mathcal{B}_2$ scheme so that the user shares the *characteristic vectors* of the index components i_1, i_2, i_3 among the two databases, in addition to its original queries. That is, for each component i_m it sends two random binary strings (one to $\mathcal{DB}_1$ and one to $\mathcal{DB}_2$) that xor to the characteristic vector χ_{i_m}. The reconstruction function can then be represented as a Boolean xor of $O(n^{1/3})$ functions, each depending on a constant number of bits (namely, functions of the form $(s_1 \oplus s_2) \wedge b_m$, where s_1 and s_2 are two corresponding shared bits held by $\mathcal{DB}_1$ and $\mathcal{DB}_2$ respectively). Such a function can be computed by a linear-size constant-width *permutation* branching program, and thus (using techniques from [11]) has linear PSM complexity.

[4] The problem of generalized secret sharing extends the usual notion of t-out-of-m secret sharing [17] in the following way. Instead of allowing every set of participants whose size is at least t to reconstruct the secret, the class of such qualified sets is defined by a monotone Boolean function (access structure) $f_M : \{0,1\}^m \to \{0,1\}$ in the natural way. The combined shares of every unqualified set of participants should give no information about the secret. See [4, 13] for a formal definition.

a secret (from the same secret domain) subject to the condition f, whose total communication complexity is c, assuming that all *k players know the secret.*

Proof. Let $P_1', \ldots, P_m'$ denote the participants in the generalized secret sharing scheme $\mathcal{S}$. The conditional disclosure protocol $\mathcal{P}$ proceeds as follows. Each player, holding input bits $(y_j)_{j\in T}$ for some $T \subseteq [n]$, uses the shared random input to compute (as specified by $\mathcal{S}$) the shares of all participants P_i' such that: (1) g_i is a function of some y_j it holds, *and* (2) $g_i(y_j) = 1$. All such shares are simultaneously sent to Carol, who uses her input y to determine whether $f_M(\vec{g}(y)) = 1$, and if so reconstructs the secret from the shares she received using the reconstruction function of $\mathcal{S}$.

The privacy and correctness of this protocol follow from observing that the m-bit characteristic vector z of the shares received by Carol satisfies $z = \vec{g}(y)$, so that $f_M(z) = f_M(\vec{g}(y)) = f(y)$. Indeed, if $f(y) = 0$ then $f_M(z) = 0$, which by the privacy of $\mathcal{S}$ implies that the shares received by Carol give her no information about the secret. If $f(y) = 1$ then $f_M(z) = 1$, which by the correctness of $\mathcal{S}$ implies that Carol can reconstruct the secret. $\square$

Corollary 1. *Suppose $f : \{0,1\}^n \rightarrow \{0,1\}$ has a Boolean formula of size $S(n)$,*[5] *and let s denote a secret bit known to* all *players. Then there exists a protocol for disclosing s subject to the condition f, whose total communication complexity is $S(n)$.*

Proof. Let F be a Boolean (AND-OR) formula for f over the literals $\{y_1, \overline{y}_1, \ldots, y_n, \overline{y}_n\}$, whose size is $S(n)$. Replacing each negative literal $\overline{y}_j$ with a positive literal w_j, we obtain a monotone Boolean formula F_M of size $S(n)$ computing a *monotone* Boolean function $f_M(y_1, \ldots, y_n, w_1, \ldots, w_n)$. Since f is a projection of f_M (as $f(y_1, \ldots, y_n) = f_M(y_1, \ldots, y_n, \overline{y}_1, \ldots, \overline{y}_n)$), then by Proposition 2 the corollary follows. $\square$

The use of shared randomness allows to obtain essentially the same result even when only one player knows the secret.

Corollary 2. *Let s denote a secret bit known to* at least one *player, and let f be as in Corollary 1. Then there exists a protocol for disclosing s subject to the condition f, whose total communication complexity is $S(n) + 1$.*

Proof. The players conditionally disclose a shared random bit r subject to the condition f (using $S(n)$ communication), and a single player holding s simultaneously sends the bit $s \oplus r$ to Carol. Clearly, if Carol can reconstruct r then she can also reconstruct s, and if she obtains no information on r then she can obtain no information on s. $\square$

An honest-user-SPIR counterpart of the k-database $\mathcal{B}_k$ scheme of [1] will be obtained by combining the use of a PSM protocol with the use of conditional disclosure of secrets. As a basis for the general recursive construction, we initially show that the 2-database $\mathcal{B}_2$ scheme can be transformed into an honest-user-SPIR scheme $\mathcal{B}_2'$ of the same asymptotic complexity.

Theorem 2. *There exists a 2-database honest-user-SPIR scheme, $\mathcal{B}_2'$, of communication complexity $O(n^{1/3})$.*

Proof. We first describe the original $\mathcal{B}_2$ scheme. This scheme may be regarded as a 2-database implementation of the 8-database 3-dimensional elementary cube scheme described in subsection 3.1. Let $\ell = n^{1/3}$, and let $i = (i_1, i_2, i_3)$ be the index of the data bit being retrieved. Each of the two databases $\mathcal{DB}_{000}$ and $\mathcal{DB}_{111}$ emulates the 4 databases $\mathcal{DB}_\sigma$ such that $\sigma \in \{0,1\}^3$ is within Hamming distance (at most) 1 from its index. This is done in the following way. The user sends to $\mathcal{DB}_{000}$ the subcube $C_{000} = (S_1^0, S_2^0, S_3^0)$ and to $\mathcal{DB}_{111}$ the subcube $C_{111} = (S_1^1, S_2^1, S_3^1)$ as in the elementary cube scheme. We would like the answers of each of the two databases to include the 4 answer bits of the 4 databases it emulates. To achieve this

[5] Using the secret sharing scheme proposed in [14], this can be generalized to any function f with a *span program* (over $GF(2)$) of size $S(n)$.

goal, $\mathcal{DB}_{000}$ sends the single bit b_{000} (which it is able to compute from the query it received) along with 3 ℓ-bit long strings, each of which contains the answer bit of one of the other databases it emulates. For instance, the i'_1-th bit of the string emulating $\mathcal{DB}_{100}$ is obtained by computing the exclusive-or of all data bits residing in the subcube $(S_1^0 \oplus i'_1, S_2^0, S_3^0)$, implying that the i_1-th bit in this string is equal to b_{100}. Symmetrically, $\mathcal{DB}_{111}$ sends the single bit b_{111} along with 3 ℓ-bit long strings, each of which corresponds to the subcubes obtained from C_{111} by xoring a single set S_m^1 with all ℓ possible values of i'_m. Altogether, the user receives 8 answer strings $a_\sigma, \sigma \in \{0,1\}^3$, six of which contain ℓ bits each, and the other two (namely, a_{000} and a_{111}) contain single bits. In each of the ℓ-bit long strings, the index of the required answer bit b_σ is either i_1 (for $\sigma = 100, 011$), i_2 ($\sigma = 010, 101$), or i_3 ($\sigma = 001, 110$). Since the user knows the positions of all 8 bits b_σ, $\sigma \in \{0,1\}^3$, in the answer strings, it can reconstruct x_i by computing the exclusive-or of these bits.

It is convenient to view the reconstruction function of the $\mathcal{B}_2$ scheme as a two-stage procedure:

1. The user selects a *single* bit from each of 8 answer strings, depending only on the index i.

2. The user exclusive-ors the 8 bits it has selected to obtain x_i.

Thus, if we let the honest user learn only the exclusive-or of the 8 bits corresponding to the index i, the data privacy requirement will be met. This can be achieved by using the conditional disclosure of secrets primitive *on top* of a PSM protocol computing the exclusive-or of 8 bits. The scheme $\mathcal{B}'_2$, an honest-user-SPIR version of $\mathcal{B}_2$, may proceed as follows:

Queries: The user sends the subcubes C_{000} to $\mathcal{DB}_{000}$ and C_{111} to $\mathcal{DB}_{111}$, as in the $\mathcal{B}_2$ scheme. In addition, the user independently shares the characteristic vectors χ_{i_m}, $m = 1, 2, 3$, among the two databases. This is done by picking random ℓ-bit strings r_m^0, r_m^1 such that $r_m^0 \oplus r_m^1 = \chi_{i_m}$ and sending the three strings r_m^0 to $\mathcal{DB}_{000}$ and the three strings r_m^1 to $\mathcal{DB}_{111}$.

Answers: Each of the two databases computes 4 answer strings as in the $\mathcal{B}_2$ scheme. Denote by a_σ the answer string emulating $\mathcal{DB}_\sigma$, $\sigma \in \{0,1\}^3$. The databases treat each bit of a string a_σ as an input to a PSM protocol computing the Boolean xor of 8 bits, and using their shared randomness they compute the PSM message sent for each such bit. Under the simple PSM protocol for xor (see subsection 3.1), each such message consists of a single bit. Let w_σ denote the string obtained by replacing each bit from a_σ by its corresponding PSM message bit. In this case, w_σ is obtained by xoring every bit of a_σ with the same random bit r_σ, where the bits r_σ are 8 random bits that xor to 0. Finally, for every $\sigma \in \{0,1\}^3$ and $1 \le j \le |w_\sigma|$, the databases use their shared randomness to disclose to the user the j-th bit of w_σ, $(w_\sigma)_j$, subject to an appropriate condition. For $\sigma = 100, 011$ the condition is $(r_1^0)_j \oplus (r_1^1)_j = 1$, for $\sigma = 010, 101$ the condition is $(r_2^0)_j \oplus (r_2^1)_j = 1$, and for $\sigma = 001, 110$ the condition is $(r_3^0)_j \oplus (r_3^1)_j = 1$. The single bits w_{000}, w_{111} can be sent in a plain form.

Reconstruction: The user reconstructs the eight PSM message bits corresponding to the index i (using the reconstruction function of the conditional disclosure protocol), and xors them to obtain x_i.

The correctness of the above scheme and the user's privacy are easy to verify. The scheme's data privacy (when the user is honest) follows from the following observations:

- By the use of conditional disclosure of secrets, the answer strings disclose to the user only 8 bits: a single bit from each w_j, whose position corresponds to the appropriate characteristic vector shared by the user.

- By the privacy of the underlying PSM protocol for "xor", the joint distribution of these 8 bits depends only on their exclusive-or, which equals x_i.

- By the independence of the PSM randomness and the conditional disclosure randomness, the user's view given its queries depends only on x_i.

(This argument can be easily formalized by describing a simulator which, given an honest user's queries and a value of x_i, produces the answer distribution).

Since the condition for disclosing each of the $O(n^{1/3})$ bits of the strings w_j is expressed by a Boolean formula of a constant size, from Corollary 2 it follows that all such bits can be conditionally disclosed with a total communication cost of $O(n^{1/3})$ bits. Altogether, the communication complexity of the scheme is $O(n^{1/3})$, as required. $\square$

The next theorem, whose proof is deferred to the Appendix, generalizes Theorem 2 to any number of databases $k \geq 2$.

Theorem 3. *For every* fixed $k \geq 2$ *there exists a k-database honest-user-SPIR scheme,* $\mathcal{B}'_k$*, of communication complexity* $O(n^{1/(2k-1)})$.

Note that the methodology of using conditional disclosure of secrets on top of a PSM protocol may be useful for handling *any* scheme whose reconstruction depends only on a proper subset of the answer bits. In particular, this applies to schemes obtained from the generic database-dominated balancing technique of [10].

4 SPIR Schemes with Respect to Dishonest Users

In this section we transform PIR schemes into SPIR schemes which guarantee data privacy with respect to dishonest users as well. The following example demonstrates the undesired extra power granted to a dishonest user in ordinary PIR schemes.

Example 1. *Consider the* $\mathcal{B}_2$ *scheme. By sending the subcube* $C = (\{i_1\}, \{i_2\}, \{i_3\})$ *(which is a legitimate query) to the 1st database, its answers alone reveal about* $3n^{1/3}$ physical *bits of data. Namely, the data bits revealed are all those whose Hamming distance from* (i_1, i_2, i_3) *is at most 1. Although an honest user might also be lucky enough to learn that many physical data bits, this occurs with only a negligible probability.*

In principle, every honest-user-SPIR scheme can be made resilient to dishonest users by filtering every original answer bit using the conditional disclosure primitive, where the condition tests for the validity of the user's queries. For instance, to properly handle the honest-user-SPIR version of $\mathcal{B}_2$ (see Theorem 2), such a condition must ensure in particular that the user does not try to obtain several bits from one string w_σ, e.g. by sharing the all-ones vector instead of a characteristic vector χ_{i_m}. However, the complexity of disclosing each answer bit subject to a full validity test will usually be prohibitive. We significantly reduce this overhead by requiring the user to send a "proof" for the validity of its queries (which may be viewed as switching to *nondeterministic* verification), and by letting different answer bits be disclosed subject to different conditions.

The next subsections present specific transformations from PIR schemes into corresponding SPIR schemes. All transformations involve at most an $O(\log n)$ multiplicative communication overhead.

4.1 Cube Schemes

In this subsection we construct a k-database SPIR scheme of complexity $O(\log n \cdot n^{1/(2k-1)})$. As in the previous section, we first address the 2-database case.

Theorem 4. *There exists a 2-database SPIR scheme,* $\mathcal{B}''_2$*, of communication complexity* $O(\log n \cdot n^{1/3})$.

Proof. Assume that $\ell = n^{1/3}$ is a power of 2. For every $j \in [\ell]$, let $\mathrm{bin}(j)$ denote the $(\log_2 \ell)$-bit binary representation of j. The scheme $\mathcal{B}''_2$ proceeds as follows:

Queries: The user sends to $\mathcal{DB}_{000}$ the subcube $C_{000} = (S^0_1, S^0_2, S^0_3)$ and to $\mathcal{DB}_{111}$ the subcube $C_{111} = (S^1_1, S^1_2, S^1_3)$, as in the $\mathcal{B}_2$ scheme. In addition, the user independently shares *binary* representations of the index components i_m, $m = 1, 2, 3$. This is done by picking random $(\log_2 \ell)$-bit strings r^0_m, r^1_m such that $r^0_m \oplus r^1_m = \mathrm{bin}(i_m)$, and sending the strings r^0_m to $\mathcal{DB}_{000}$ and the strings r^1_m to $\mathcal{DB}_{111}$.

Answers: Let w_σ denote the PSM message strings as in the $\mathcal{B}_2'$ scheme, and let s be a shared random bit. The databases use their shared randomness to independently disclose to the user each of the following bits:

1. The bit s is disclosed subject to the condition $\bigwedge_{m=1}^{3}(S_m^0 \oplus S_m^1 = \{r_m^0 \oplus r_m^1\})$ (which validates the user's queries). This condition can be verified by a Boolean formula of size $O(\ell \log \ell)$.[6]
2. The bits $w_{000} \oplus s$ and w_{111} are sent in a plain form. The former bit may be viewed as the PSM message corresponding to the input bit $b_{000} \oplus s$.
3. Each bit of w_σ ($\sigma \neq 000, 111$) is disclosed subject to the appropriate comparison of its position to the shared index components; e.g., the j-th bits of w_{001} and w_{110}, $1 \leq j \leq \ell$, are disclosed subject to the condition $r_3^0 \oplus r_3^1 = \mathrm{bin}(j)$. Each such condition can be verified by a Boolean formula of size $O(\log \ell)$.

Reconstruction: The honest user can reconstruct s and the 8 bits corresponding to the index i. The retrieved bit x_i is equal to the exclusive-or of these 9 bits.

The correctness and the user's privacy in the above scheme are easy to verify. The scheme's data privacy, relative to *any* user, follows from the following observations:

- From each of the 6 ℓ-bit strings w_σ the user can obtain only a *single physical bit* of w_σ, corresponding to the appropriate shared index component. By the use of an underlying PSM protocol for xor, this means that the user can only learn $(s \oplus b_{000}) \oplus b_{111} \oplus b$, where b is the exclusive-or of the 6 selected bits.
- If the user's queries are inconsistent, i.e. the relation between the two subcubes does not match the shared index components, then the user obtains no information on s, and hence (by the previous observation) no information at all. On the other hand, if the user's queries are consistent with some index i, then it learns only s and $s \oplus x_i$, which is equivalent to learning x_i.

From the sizes of the Boolean formulas used as disclosure conditions it follows that the scheme meets the specified complexity bound. □

Theorem 4 is generalized by the following theorem, whose proof appears in the Appendix.

Theorem 5. *For every* fixed $k \geq 2$ *there exists a k-database SPIR scheme, $\mathcal{B}_k''$, of communication complexity* $O(\log n \cdot n^{1/(2k-1)})$.

4.2 A Polynomial Interpolation Scheme

In this subsection we show how the polynomial interpolation PIR scheme for $k = \log_2 n + 1$ databases [10] (see also [2]) can be transformed into a SPIR scheme with the same number of databases, and with an $O(\log \log n)$ factor of communication overhead.

Theorem 6. *There exists a* $\lceil \log_2 n + 1 \rceil$*-database SPIR scheme of communication complexity* $O(\log^2 n \cdot (\log \log n)^2)$.

Proof. We start by describing the elementary polynomial interpolation scheme. Suppose $n = 2^s$, and let $k = s + 1$ be the number of databases. Let $GF(q)$ be a finite field with at least $s + 2$ elements, and α_j, $1 \leq j \leq k$, be distinct, nonzero elements of $GF(q)$. With every index $i \in [n]$ we associate an s-tuple $\vec{i} = (i_1, i_2, \ldots, i_s) \in \{0,1\}^s$, corresponding to the binary representation of i. For each data string $x \in \{0,1\}^n$ there exists a multivariate degree-s polynomial $p_x(y_1, \ldots, y_s)$, such that $p_x(\vec{i}) = x_i$ for every $i \in [n]$. The user picks a random s-tuple $\vec{c} = (c_1, \ldots, c_s) \in GF(q)^s$, and sends to each database $\mathcal{DB}_j$,

[6] A condition of the form $S \oplus S' = \{r \oplus r'\}$ can be expressed as a conjunction of ℓ conditions of the form $((S)_j \oplus (S')_j = 1) \leftrightarrow (r \oplus r' = \mathrm{bin}(j))$, where each such smaller condition can be easily verified by a formula of size $O(\log \ell)$.

$1 \le j \le k$, the query $\vec{w}^j = \alpha_j \cdot \vec{c} + \vec{i}$.[7] Each database $\mathcal{DB}_j$ replies with a single field element $a_j \triangleq p_x(\vec{w}^j)$. The user reconstructs x_i by interpolation: if p' is the unique degree-s univariate polynomial (over $GF(q)$) such that $p'(\alpha_j) = a_j$ for every $1 \le j \le k$, then $x_i = p'(0)$. The communication complexity of the scheme is $O(\log^2 n \log\log n)$.

In Section 3 we relied on the linearity of the reconstruction function to obtain a PSM-based honest-user-SPIR scheme with the same communication complexity. To prevent a dishonest user from obtaining any illegitimate information on x, we require the user to prove that its queries are consistent with some $\vec{i} \in \{0,1\}^s$ and $\vec{c} \in GF(q)^s$. Such a proof will consist of sharing each entry of $\vec{c}$ and $\vec{i}$, and its validation will consist of verifying that $\vec{i} \in \{0,1\}^s$ and that $\vec{w}^j = \alpha_j \cdot \vec{c} + \vec{i}$ for each $1 \le j \le k$. The SPIR scheme is formally described in the following.

Queries: The user sends to each database $\mathcal{DB}_j$ a query $\vec{w}^j$ as in the original scheme. In addition, the user picks random tuples $\vec{i}^0, \vec{i}^1, \vec{c}^0, \vec{c}^1 \in GF(q)^s$ such that $\vec{i}^0 + \vec{i}^1 = \vec{i}$ and $\vec{c}^0 + \vec{c}^1 = \vec{c}$, and sends $\vec{i}^0, \vec{c}^0$ to $\mathcal{DB}_1$ and $\vec{i}^1, \vec{c}^1$ to each of $\mathcal{DB}_2, \ldots, \mathcal{DB}_k$.

Answers: By linearity of the original scheme's reconstruction function, it can be expressed as $\sum_{j=1}^k \beta_j a_j$, where the β_j's are fixed field elements (which depend on the α_j's). Let $r_1, r_2, \ldots, r_k$ be independent, random elements of $GF(q)$ (included in the databases' shared randomness), and let $r = -\sum_{j=1}^k r_j$. Each database $\mathcal{DB}_j$ replies with $a'_j \triangleq \beta_j a_j + r_j$, where a_j is its answer according to the original scheme. In addition, the databases use their shared randomness to independently disclose each bit of r, subject to the following conditions: (1) for every $3 \le j \le k$, the shares of $\vec{i}$ and $\vec{c}$ sent to $\mathcal{DB}_j$ must be identical to those sent to $\mathcal{DB}_2$; (2) for every $1 \le m \le s$, either $i^0_m + i^1_m = 0$ or $i^0_m + i^1_m = 1$ (where i^b_m denotes the m-th entry of the b-th share of $\vec{i}$); and finally: (3) for every $1 \le j \le k$ and $1 \le m \le s$, we require that $\alpha_j(c^0_m + c^1_m) + (i^0_m + i^1_m) = w^j_m$. Note that each of the atomic conditions can be expressed as equality between two elements of $GF(q)$, known to two different databases. For instance, if $j > 1$ then verifying the condition $\alpha_j(c^0_m + c^1_m) + (i^0_m + i^1_m) = w^j_m$ is equivalent to comparing $\alpha_j c^0_m + i^0_m$, which is known to $\mathcal{DB}_1$, and $w^j_m - \alpha_j c^1_m - i^1_m$, which is known to $\mathcal{DB}_j$. Thus each such atomic condition can be verified by a Boolean formula of size $O(\log s)$, and the whole disclosure condition by a Boolean formula of size $O(s^2 \log s)$. Altogether, disclosing all bits of r requires $O(s^2 \log^2 s) = O(\log^2 n (\log\log n)^2)$ communication bits.

Reconstruction: The user reconstructs r, and computes x_i as $\sum_{j=1}^k a'_j + r$.

The correctness and the user's privacy of the original scheme are clearly maintained. To see the data privacy of this scheme, consider two possible cases. If the user's queries are valid, then the tuple $(a'_1, a'_2, \ldots, a'_k, r)$ is uniformly distributed among all $(k+1)$-tuples over $GF(q)$ which sum up to x_i, implying that the answer distribution depends only on x_i. Otherwise, the user obtains no information on r, and consequently $a'_1, \ldots, a'_k$ (which are uniformly and independently distributed over $GF(q)$) are independent of the conditional disclosure messages. It follows that in the latter case the user learns nothing at all.

The communication complexity is dominated by the conditional disclosure of r, which requires $O(\log^2 n \cdot (\log\log n)^2)$ communication bits. □

4.3 Computational PIR

In a computational setting, where requirements of strict equality between distributions are relaxed to computational indistinguishability requirements, our information-theoretic primitives may be replaced by more powerful cryptographic primitives. In [11] it is shown that for every function in P there exists a *computational* PSM protocol of polynomial communication complexity, assuming the existence of a one-way function. Since both the reconstruction function and the validity of the user's queries in the computational scheme of [9] can be verified in polynomial time (uniformly in c), we may conclude the following:

[7] Throughout this subsection, all additions and multiplications are over $GF(q)$.

Theorem 7. *There exists a 2-database* computational *SPIR scheme of communication complexity $O(n^c)$ (for any $c > 0$, assuming the existence of a one-way function).*

5 Final Remarks

From a wider perspective, our solutions exploit the specific structure of current PIR schemes (single round, a simple-enough structure of reconstruction function) to provide efficiency which cannot be rivaled in more general settings. For instance, general techniques of information-theoretic resilient multi-party computation (cf. [3, 8]), which can in principle be applied to convert PIR schemes into SPIR counterparts, involve a much higher complexity overhead.

The techniques we have used can be applied to obtain SPIR analogues of other PIR-related results which were not discussed in this work. In particular, efficient SPIR schemes for retrieving *blocks* of data, and schemes with a higher degree of user privacy (i.e., privacy with respect to any coalition of at most t databases) may be obtained by applying similar techniques to schemes from [10]. In the case of *single*-database computational PIR schemes [15], (multi-round) SPIR counterparts can be obtained by applying standard zero-knowledge techniques (whose polynomial communication overhead is insignificant in that case).

We finally note that the SPIR problem may be viewed as a distributed version of the problem of $\binom{n}{1}$-Oblivious-Transfer [6, 7] (also denoted "all or nothing disclosure of secrets"). Thus, our solutions immediately give 1-round distributed implementations of $\binom{n}{1}$-Oblivious-Transfer with sub-linear communication complexity.

References

[1] A. Ambainis. Upper bound on the communication complexity of private information retrieval. In *Proc. of ICALP '97, to appear.*

[2] D. Beaver and J. Feigenbaum. Hiding instances in multioracle queries. In *STACS*, 1990.

[3] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proc. of 20th STOC*, pages 1–10, 1988.

[4] J. Benaloh and J. Leichter. Generalized secret sharing and monotone functions. In *Proc. of CRYPTO '88*, pages 27–35, 1990.

[5] M. Blum, P. Feldman, and S. Micali. Non-interactive zero-knowledge and its applications. In *Proc. of 20th STOC*, pages 103–112, 1988.

[6] G. Brassard, C. Crépeau, and J.-M. Robert. Information theoretic reductions among disclosure problems. In *Proc. of 18th STOC*, pages 168–173, 1986.

[7] G. Brassard, C. Crépeau, and M. Santha. Oblivious transfers and intersecting codes. In *IEEE Transaction on Information Theory, special issue on coding and complexity.* November 1996.

[8] D. Chaum, C. Crépeau, and I. Damgård. Multiparty unconditionally secure protocols (extended abstract). In *Proc. of 20th STOC*, pages 11–19, 1988.

[9] B. Chor and N. Gilboa. Computationally private information retrieval. In *Proc. of 29th STOC, to appear*, 1997.

[10] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In *Proc. of 36th FOCS*, pages 41–50, 1995.

[11] U. Feige, J. Kilian, and M. Naor. A minimal model for secure computation (extended abstract). In *Proc. of 26th STOC*, pages 554–563, 1994.

[12] Y. Ishai and E. Kushilevitz. Private simultaneous messages protocols with applications. In *Proc. of 5th ISTCS, to appear*, 1997.

[13] M. Ito, A. Saito, and T. Nishizeki. Secret sharing schemes realizing general access structures. In *Proc. IEEE Global Telecommunication Conf., Globecom 87*, pages 99–102, 1987.

[14] M. Karchmer and A. Wigderson. On span programs. In *Proc. of 8th IEEE Structure in Complexity Theory*, pages 102–111, 1993.

[15] E. Kushilevitz and R. Ostrovsky. Single-database computationally private information retrieval. Submitted to these proceedings.

[16] R. Ostrovsky and V. Shoup. Private information storage. In *Proc. of 29th STOC, to appear*, 1997.

[17] A. Shamir. How to share a secret. *Commun. ACM*, 22(6):612–613, June 1979.

[18] S. Skyum and L. Valiant. A complexity theory based on Boolean algebra. *JACM*, 32:484–502, 1985.

A Necessity of Shared Randomness

In this section we show that the addition of a shared randomness resource to the basic PIR setting is in a sense minimal.

Suppose we allow the databases to use *private* randomness in answering the user's queries, but we still do not allow them to interact without the mediation of the user (and in particular we do not allow them to share a random string unknown to the user). We argue that in this setting, (information-theoretic) SPIR cannot be implemented at all, even when the user is honest.

Proposition 3. *There exists no multi-round k-database SPIR scheme without interaction between the databases, even if the databases are allowed to hold* private, *independent random inputs, and the user is honest.*

Proof. The strong privacy requirement implies that any single database $\mathcal{DB}_j$ cannot respond to the user's queries in a way that depends on the data string x. Formally, at any round the distribution of the database's answer given previous communication is the same under every x. For otherwise, this answer distribution must either not follow from x_i alone, thus violating the data's privacy, or alternatively reveal the index i on which it depends, thus violating the user's privacy. By independence of private random inputs held by different databases, this implies that the *joint* distribution of their answers, given previous communication, is independent of x. Fixing an index i, it follows by induction on the number of rounds that for any $w > 0$ the accumulated communication in the first w rounds is distributed independently of x. In particular, this implies that the user's output cannot depend on the value of x_i, contradicting the correctness requirement. $\square$

B Proofs

Proof of Theorem 3. We show how to adapt the proof of Theorem 2 to the k-database generalization given in [1]. Let $d = 2k - 1$ and $\ell = n^{1/d}$. In the $\mathcal{B}_k$ scheme, the k databases (denoted $\mathcal{DB}_1, \ldots, \mathcal{DB}_k$) jointly emulate the 2^d databases of the d-dimensional elementary cube scheme. The scheme proceeds as follows. The user sends to $\mathcal{DB}_1$ the subcube C_{0^d} as in the elementary cube scheme, and sends to each of $\mathcal{DB}_2, \ldots, \mathcal{DB}_k$ the subcube C_{1^d}. In its answers, $\mathcal{DB}_1$ emulates all databases $\mathcal{DB}_\sigma$ of the original scheme such that $\sigma \in \{0,1\}^d$ is at Hamming distance at most 1 from 0^d, similarly to the way such an emulation is done in the $\mathcal{B}_2$ scheme. Simultaneously, the remaining databases $\mathcal{DB}_2, \ldots, \mathcal{DB}_k$ jointly emulate the remaining databases of the original schemes, namely all $\mathcal{DB}_\sigma$ such that σ contains at least two 1's. This is done using a constant number $(2^d - d - 1)$ of recursive invocations of the $\mathcal{B}_{k-1}$ scheme between the user and $\mathcal{DB}_2, \ldots, \mathcal{DB}_k$. In each such invocation the user retrieves a single bit b_σ from a virtual data string, whose entries correspond to the different subcubes possibly sent to $\mathcal{DB}_\sigma$ in the elementary scheme (i.e.., each bit of the virtual data strings equals the exclusive-or of data bits residing in such a potential subcube). By xoring $d+1$ bits selected from the answers of $\mathcal{DB}_1$ together with the $2^d - d - 1$ bits retrieved by the recursive invocations of $\mathcal{B}_{k-1}$, the user reconstructs x_i.

It is not hard to see that the technique we used for converting $\mathcal{B}_2$ into an honest-user-SPIR scheme $\mathcal{B}'_2$ of the same asymptotic complexity can be carried on recursively. Suppose we have a $(k-1)$-database honest-user-SPIR scheme $\mathcal{B}'_{k-1}$ of communication complexity $O(n^{1/(2k-3)})$. We can obtain a recursive definition of $\mathcal{B}'_k$ (in terms of $\mathcal{B}'_{k-1}$) from the recursive definition of $\mathcal{B}_k$, similarly to the way $\mathcal{B}'_2$ was obtained from $\mathcal{B}_2$. Omitting details which should be clear by now, $\mathcal{B}'_k$ proceeds as follows:

- The user sends the subcube C_{0^d} to $\mathcal{DB}_1$ and the subcube C_{1^d} to each of $\mathcal{DB}_2, \ldots, \mathcal{DB}_k$. Independently, the user shares all characteristic vectors χ_{i_m}, $m = 1, \ldots, d$, between $\mathcal{DB}_1$ and $\mathcal{DB}_2$. In their answers, $\mathcal{DB}_1, \mathcal{DB}_2$ disclose each bit of the PSM message strings w_σ held by $\mathcal{DB}_1$, subject to an appropriate condition on corresponding bits of the shares they received (as in the $\mathcal{B}'_2$ scheme).

- The user and databases $\mathcal{DB}_2, \ldots, \mathcal{DB}_k$ recursively invoke $\mathcal{B}'_{k-1}$, as in the definition of $\mathcal{B}_k$, except that the virtual data strings a_σ used in $\mathcal{B}_k$ are replaced by the corresponding PSM message strings w_σ. I.e., every bit of a_σ (from which the bit b_σ is retrieved) is xored with the same shared random bit r_σ.

Remarks:

- The above scheme requires only one round of interaction. Since the conditional disclosure of each bit requires only a constant amount of communication, the communication complexity satisfies: $c_k(n) \leq (d+1) \cdot O(\ell) + (2^d - d - 1) \cdot c_{k-1}(\ell^{d-2}) = O(n^{1/(2k-1)})$.

- Databases $\mathcal{DB}_2, \ldots, \mathcal{DB}_k$ do not have to *directly* apply the conditional disclosure primitive; the recursive use of $\mathcal{B}'_{k-1}$ "takes care" of exposing only a single bit b_σ per invocation.

- The scheme $\mathcal{B}'_2$ may serve as basis for the recursion (in contrast to the original PIR setting, in which it was possible to use a trivial scheme $\mathcal{B}_1$ as basis).

$\square$

Proof of Theorem 5. As in the proof of Theorem 3, we follow the recursive construction of [1]. Let $d = 2k - 1$ and $\ell = n^{1/d}$. Suppose we have a $(k-1)$-database SPIR scheme $\mathcal{B}''_{k-1}$ of communication complexity $O(\log n \cdot n^{1/(2k-3)})$. In this case we make an additional assumption on $\mathcal{B}''_{k-1}$: we assume that the user is required to *commit* to the index being retrieved. This assumption, referred to as a *strong data-privacy requirement* is formally defined in the following.

We say that a 1-round PIR scheme $\mathcal{P}$ satisfies the *strong data-privacy requirement* with parameter d', if the following conditions hold:

1. On a data string x of length $n' = \ell^{d'}$, the user sends special queries Q^0_m, Q^1_m, $m \leq 1 \leq d'$ (each of which is a $(\log_2 \ell)$-bit string); and

2. If a user (possibly a dishonest user) sends queries in which $Q^0_m \oplus Q^1_m = \mathrm{bin}(i'_m)$ for each $1 \leq m \leq d'$, then the answers reveal at most the bit $x_{(i'_1,\ldots,i'_{d'})}$.

Note that strong data privacy implies the usual data privacy. Also note that the scheme $\mathcal{B}''_2$ satisfies this stronger requirement with $d' = 3$, since in that scheme the user can only obtain information on the data bit corresponding to index components i_1, i_2, i_3 it shares between the two databases. Our additional assumption on $\mathcal{B}''_{k-1}$ (which will be carried on to $\mathcal{B}''_k$) is that it satisfies the strong data privacy requirement with $d' = 2(k-1) - 1 = 2k - 3$.

The scheme $\mathcal{B}''_k$ proceeds as follows:

- The user sends the subcube $C_{0^d} = (S^0_1, \ldots, S^0_d)$ to $\mathcal{DB}_1$ and the subcube $C_{1^d} = (S^1_1, \ldots, S^1_d)$ to each of $\mathcal{DB}_2, \ldots, \mathcal{DB}_k$. Independently, the user shares *binary* representations of the index components i_m, $m = 1, 2, \ldots, d$ (as in the $\mathcal{B}''_2$ scheme). In their answers, $\mathcal{DB}_1, \mathcal{DB}_2$ disclose each j-th bit of a PSM message string w_{e_m}, $1 \leq j \leq \ell, 1 \leq m \leq d$, subject to the condition $i_m = j$ (where e_m denotes the m-th unit vector of length d). In addition, $\mathcal{DB}_1$ sends the bit $w_{0^d} \oplus s$, where s is a shared random bit.

- For each $\sigma \in \{0,1\}^d$ such that $\mathrm{weight}(\sigma) \geq 2$, the user and the databases $\mathcal{DB}_2, \ldots, \mathcal{DB}_k$ recursively invoke $\mathcal{B}''_{k-1}$ on the virtual data string w_σ defined in the following. Let $d' = d-2$ and $n' = \ell^{d'}$. Let m^σ_z, $1 \leq z \leq \mathrm{weight}(\sigma)$, denote the position of the z-th zero in σ. With every σ such that $\mathrm{weight}(\sigma) \geq 2$ and tuple $i' = (i'_1, \ldots, i'_{d'}) \in [\ell]^{d'}$ we associate a subcube $C^\sigma_{i'}$ (of the cube $[\ell]^d$), which is obtained from C_{1^d} by replacing each set S^1_z, $1 \leq z \leq \mathrm{weight}(\sigma)$, with the set $S^1_z \oplus i_{m^\sigma_z}$. Each w_σ is defined to be the n'-bit string, whose i'-th bit is equal to the exclusive-or of data bits residing in the subcube $C^\sigma_{i'}$ together with the PSM random bit r_σ. In a recursive invocation of $\mathcal{B}''_{k-1}$ on the virtual data string w_σ, the user retrieves the bit whose index is represented by the d'-tuple $i'_\sigma = (i_{m^\sigma_1}, i_{m^\sigma_2}, i_{m^\sigma_p}, 1, \ldots, 1)$, where $p = \mathrm{weight}(\sigma)$.

- The shared random bit s is disclosed subject to a conjunction of the following conditions:

  1. For every $3 \leq j \leq k$, the subcube sent to $\mathcal{DB}_j$ is equal to the subcube sent to $\mathcal{DB}_2$.
  2. The subcubes sent to $\mathcal{DB}_1, \mathcal{DB}_2$ are consistent with the components of the index i shared between these two databases (see $\mathcal{B}_2''$ for implementation details).
  3. For every $\sigma \in \{0,1\}^d$ such that weight$(\sigma) \geq 2$, the index i' shared by the user in the invocation of $\mathcal{B}_{k-1}''$ on w_σ (in accordance with the strong data privacy assumption made on $\mathcal{B}_{k-1}''$) is equal to i'_σ. This can be verified by comparing each component of i' with the corresponding component of i as shared by the user.

We start by analyzing the communication complexity. The combined size of all conditional disclosure formulas is $O(\ell \log \ell)$. Thus, the communication complexity satisfies $c_k(n) = O(\ell \log \ell) + (2^d - d - 1) \cdot c_{k-1}(\ell^{d-2}) = O(\ell \log \ell) = O(\log n \cdot n^{1/(2k-1)})$.

The correctness and the user's privacy are straightforward to verify. It remains to show that the strong data privacy requirement also holds for $\mathcal{B}_k''$. We argue that if the user commits to an index $i = (i_1, \ldots, i_d)$ (by sharing its components between $\mathcal{DB}_1$ and $\mathcal{DB}_2$), then it can learn at most the bit x_i. As in the $\mathcal{B}_k'$ scheme, an *honest* user learns x_i alone. In order to learn something on other bits, a dishonest user must deviate from the scheme's specification either by sending to $\mathcal{DB}_1, \ldots, \mathcal{DB}_k$ subcubes which don't meet the requirements imposed by i, or by trying to retrieve from the recursive invocations of $\mathcal{B}_{k-1}''$ different bits than those corresponding to i. The specified disclosure conditions and the strong data privacy assumption made on $\mathcal{B}_{k-1}''$ guarantee that in both of these cases, the user will learn nothing about s. By the use of the underlying PSM protocol and by the data-privacy of $\mathcal{B}_{k-1}''$, the answers to any choice of queries reveal at most the exclusive-or of 2^d bits (one selected from each list) together with s. Thus, learning nothing about s makes the user learn nothing at all. $\square$