Single Database Private Information Retrieval Implies Oblivious Transfer *

Giovanni Di Crescenzo[†] Tal Malkin[‡] Rafail Ostrovsky[§]

Abstract

A Single-Database Private Information Retrieval (PIR) is a protocol that allows a user to privately retrieve from a database an entry with as small as possible communication complexity. We call a PIR protocol *non-trivial* if its total communication is strictly less than the size of the database. Non-trivial PIR is an important cryptographic primitive with many applications. Thus, understanding which assumptions are necessary for implementing such a primitive is an important task, although (so far) not a well-understood one. In this paper we show that any non-trivial PIR implies Oblivious Transfer, a far better understood primitive. Our result not only significantly clarifies our understanding of any non-trivial PIR protocol, but also yields the following consequences:

- Any non-trivial PIR is *complete* for all two-party and multi-party secure computations.
- There exists a communication-efficient reduction from any PIR protocol to a 1-out-of-n Oblivious Transfer protocol (also called SPIR).
- There is strong evidence that the assumption of the existence of a one-way function is necessary but not sufficient for any non-trivial PIR protocol.

*B. Preneel (ED.): EUROCRYPT 2000, LNCS 1807, pp. 122-138, 2000. Springer-Verlag.

†Telcordia Technologies, Inc., 445 South Street, Morristown, NJ, 07960. E-mail: giovanni@research.telcordia.com.

‡AT&T Labs – Research, 180 Park Ave., Florham Park, NJ, 0793A2. E-mail: tal@research.att.com. Work done at the Massachusetts Institute of Technology.

§Telcordia Technologies, Inc., 445 South Street, Morristown, NJ, 07960. E-mail: rafail@research.telcordia.com.

1 Introduction

RELATIONSHIPS BETWEEN CRYPTOGRAPHIC PRIMITIVES. One of the central questions in cryptography is to study which assumptions (if any) are necessary to implement a cryptographic protocol or task. For most primitives this answer is well understood, and falls in two categories: either one-way functions are necessary and sufficient, or stronger assumptions are necessary (i.e., one-way functions with some additional properties like trapdoor may be required). For example, pseudo-random generators [20], signature schemes [32, 36], commitment schemes [20, 30] and zero-knowledge proofs for NP [20, 30, 18, 34] are all equivalent to the existence of a one-way function. On the other hand there is a class of primitives that probably needs additional assumptions, including, for example, public-key cryptosystems, key-exchange, oblivious transfer [22], non-interactive zero-knowledge proofs of knowledge for NP [11] and any non-trivial secure two-party [4] and multi-party function evaluation [25]. Single Database *Private Information Retrieval* has received a lot of attention in the literature, however its place in the above setting was not understood. In this paper we address (and resolve) its position.

PRIVATE INFORMATION RETRIEVAL. A Private Information Retrieval (PIR) scheme allows a user to retrieve information from a database while maintaining the query private from the database managers. More formally, the database is modeled as an n-bit string x out of which the user retrieves the i-th bit x_i, while giving the database no information about the index i. The communication complexity of such a scheme is denoted by $c(n)$. A trivial PIR scheme consists of sending the entire data string to the user (i.e. $c(n)=n$), thus satisfying the PIR privacy requirement in the information-theoretic sense. We call any PIR protocol with $c(n)<n$ *non-trivial.* The problem of constructing non-trivial PIR was originally introduced by Chor et al. [8] and further studied in [8, 1, 7, 33, 27, 29, 3, 12, 16, 15, 6, 23, 28]. In [8] this problem was studied in the setting of multiple non-communicating copies of the database (further improvements were given in [1, 23]). That is, [8] show that if there are at least two or more copies of the database, then non-trivial PIR (for example, with two copies of the database, with communication complexity $c(n)=O(n^{1/3})$) is indeed possible. In the original work [8] also show that it is information-theoretically impossible to achieve a non-trivial PIR with a single copy of the database. Kushilevitz and Ostrovsky [27] have shown a way to get around this impossibility result using computational assumptions[1]. In particular, [27] show that assuming that the quadratic residuosity (number-theoretic) problem is hard, they can get Single-Database PIR protocol with $c(n) < n^{\epsilon}$ for any $\epsilon > 0$. Further constructions of single-database PIR schemes, improving either the communication or the assumption, followed [29, 37, 6, 28]. In particular, Cachin et al. [6] construct PIR with polylogarithmic communication complexity, under the so-called Φ-hiding (number-theoretic) assumption. This is essentially optimal communication complexity since the security parameter needs to

[1] Also, [7, 33] consider the use of computational assumptions in the settings of multiple non-communicating databases.

be at least poly-logarithmic in n. Recently, [28] have shown a single database PIR based on any one-way trapdoor permutation, though their communication, while less then n, is bigger than schemes based on specific number-theoretic assumptions [27, 29, 37, 6]. On the other hand, [3] have shown that any non-trivial single database PIR implies the existence of a one-way function.

Oblivious Transfer. The Oblivious Transfer (OT) protocol was introduced by Rabin [35], one-out-of-two Oblivious Transfer, denoted $\binom{2}{1}$-OT , was introduced in [13], and one-out-of-n Oblivious Transfer, denoted $\binom{n}{1}$-OT , was introduced in [2]. All these OT variants were shown to be equivalent to one another [10, 2]. In this paper, we will mainly use the last two versions. Roughly speaking, $\binom{2}{1}$-OT is a protocol between two players, a sender Alice and a receiver Bob. Alice has two bits, and Bob wishes to get one of them such that (a) Alice does not know which bit Bob got; and (b) Bob does not learn any information about the bit that he did not get. When generalized to $\binom{n}{1}$-OT we can see that the formulation of this primitive is "close" to single-database PIR, in that they both share requirement (a). However, non-trivial PIR has *an additional* requirement regarding the communication complexity (to be less than the number of bits) and *does not require* condition (b) – which is essential for the definition of Oblivious Transfer. The $\binom{n}{1}$-OT protocol that combines both requirements (a), (b) and the small communication requirement was considered in [16, 27], who call it Symmetric-PIR.

In [24], it was shown that OT is complete, namely it can be used to construct any other protocol problem. [21] have shown that OT implies the existence of one-way functions. Moreover, [22] have shown that assuming OT is probably stronger than assuming existence of one-way functions (OWF) in the following sense. They show that it is impossible to construct a black-box reduction from OT to OWF (where the OT protocol uses the promised OWF as a black box, and the proof is black-box). Furthermore, proving any such black-box construction (even if the proof itself is not black-box), is as hard as separating $\mathcal{P}$ from $\mathcal{NP}$. Thus [22] gives a strong evidence that OWF are currently not sufficient to construct OT, namely that OT is a strictly stronger assumption.

Our Results

In this paper, we present a reduction transforming any nontrivial single-database PIR into Oblivious Transfer. The significance of this reduction is threefold: (1) it provides "negative" results, asserting that PIR cannot be constructed based on weak computational assumptions; (2) It provides a general "positive" result, namely that PIR is also a complete primitive, and any non-trivial implementation of Single-Database PIR may be used to construct any other secure protocol; and (3) it provides a specific "positive" result, allowing transformation from communication efficient single-database PIR to communication-efficient $\binom{n}{1}$-OT (also called Symmetric-PIR [16, 27]). We elaborate below.

Complexity of PIR. As mentioned above, the original paper of Chor et al. [8] shows that it is information-theoretically impossible to implement a non-trivial Single-Database PIR. That is, if the user needs information-theoretic privacy, the communication cannot be less than n. Thus, *some* computational assumption is necessary. Naturally, this leads to the following question.

> *Under which computational assumptions can non-trivial Single-Database PIR be achieved?*

While this question has received a lot of attention recently [27, 29, 37, 6, 3, 28], only limited progress has been achieved thus far towards a solution. In particular, as described above, there has been a large gap between the assumptions known to be *sufficient*, and those known to be *necessary*. On one hand, the only assumption previously known to be necessary for non-trivial PIR is the existence of one-way functions [3]; on the other hand, the weakest assumptions known to be sufficient are trapdoor permutations [28]. In this paper we make an important step towards closing this gap, by showing the following

> **Main Theorem (Informal Statement)** *If there exists any non-trivial Single-Database PIR then there exists an OT.*

That is, even saving one bit compared to the (information-theoretic) trivial protocol of sending the entire database, already requires OT. It is interesting to note that we can also reduce any code for non-trivial single-database PIR to a code for OT; this is similar to code-to-code reductions in [4]. Moreover, our theorem holds even if the communication sent by the user in the given PIR scheme is unbounded, as long as the database sends less than n bits.

OT protocol implies the existence of a one-way function [21]. Single database PIR also implies the existence of a one-way function [3], but in light of [22] our result is strictly stronger and implies the following:

> **Corollary (Informal Statement)** *One-way functions are necessary but probably not sufficient to construct non-trivial Single-Database PIR.*

Completeness of Any non-trivial Single-Database PIR. The following corollary, demonstrating the importance of the PIR primitive, follows from the result of the completeness of OT [24]:

> **Corollary (Informal Statement)** *Any non-trivial Single-Database PIR is complete for all two-party and multi-party secure computation.*

That is, an implementation of the PIR primitives allows a secure computation of *any* function.

Symmetric-PIR (or communication-efficient $\binom{n}{1}$-OT). In the standard formulation of PIR, there is no concern about how many bits of the database the user learns. If one makes an additional requirement that the user must learn only one bit (or secret) of the database, then this can be viewed as communication-efficient $\binom{n}{1}$-OT (called Symmetrically Private Information Retrieval (SPIR)). SPIR schemes were first introduced in [16] in the setting of multiple databases. In [27] SPIR were shown to exist in the setting of a single database. The single-database SPIR schemes of [27, 16, 37] were based on specific algebraic assumptions. Naor and Pinkas [31] have shown a general reduction transforming any single database PIR into single-database SPIR using one call to the underlying PIR protocol, a logarithmic number of calls to one-out-of-two (string) Oblivious Transfer, and the existence of pseudo-random generators. Combining our main result with that of [31] we get:

> **Theorem (Informal Statement)** *If there exists any non-trivial Single-Database PIR scheme with communication $c(n)$ and security parameter k, then there exists $\binom{n}{1}$-OT (i.e., SPIR) with communication $c(n) + \mathrm{poly}(k)$.*

We stress that the efficient communication complexity of the SPIR scheme we construct is the main point of the last theorem. Indeed, in the context of computational assumptions, SPIR is equivalent to the $\binom{n}{1}$-OT variant of Oblivious Transfer. However, this theorem provides a stronger result, since the communication complexity obtained (which is the main parameter in the SPIR context) is efficient, costing only an additive factor depending on the security parameter (not on n) over the underlying PIR.

Proof Outline. The variant of OT that we use here is the $\binom{2}{1}$-OT . We prove our results using the following three steps: (1) communication-efficient PIR implies $\binom{2}{1}$-OT for honest parties; (2) communication-efficient PIR implies $\binom{2}{1}$-OT (for possibly dishonest parties); (3) communication-efficient PIR implies communication-efficient SPIR.

2 Preliminaries and Definitions

In this section we give some general conventions that we will use in the paper and the formal definitions for PIR, SPIR, and OT.

General Conventions Let $\mathbb{N}$ be the set of natural numbers and define $[k] = \{1, \ldots, k\}$. If S is a set, the notation $x \leftarrow S$ denotes the *random process* of selecting element x from set S with uniform probability distribution over S and independently from all other random choices. If A is an algorithm, the notation $y \leftarrow A(x)$ denotes the random process of obtaining y when running algorithm A on input x, where the probability space is given by uniformly and independently choosing the random coins (if any) of algorithm A. By $\mathrm{Prob}[\,R_1; \ldots; R_n : E\,]$ we denote the probability of event E, after the execution of random processes $R_1, \ldots, R_n$. We denote a distribution D as $\{R_1; \ldots; R_m : v\}$, where v denotes the values that D can assume, and $R_1, \ldots, R_m$ is a sequence of random processes generating value v. By *algorithm* we refer to a (probabilistic) Turing machine. An *interactive Turing machine* is a probabilistic Turing machine with a communication tape. A pair (A, B) of interactive Turing machines running in probabilistic polynomial time is an *interactive protocol.* A *transcript* of an execution of an interactive protocol is the sequence of messages that appear on the communication tapes of the two machines forming the protocol during that execution. The notation $t_{A,B}(x, r_A, y, r_B)$ denotes the transcript of an execution of an interactive protocol (A, B) with inputs x for A and y for B and with random strings r_A for A and r_B for B. If $t = t_{A,B}(x, r_A, y, r_B)$ is such a transcript, the output of A (resp. B) on this execution is denoted by $A(x, r_A, t)$ (resp. $B(y, r_B, t)$). The notation $(r_B, t) \leftarrow t_{A,B}(x, r_A, y, \cdot)$ denotes the random process of selecting a random string r_B uniformly at random (and independently of all other choices), and setting $t = t_{A,B}(x, r_A, y, r_B)$. Similarly we denote $(r_A, t) \leftarrow t_{A,B}(x, \cdot, y, r_B)$ for the case where A's random string is chosen uniformly at random, and $(r_A, r_B, t) \leftarrow t_{A,B}(x, \cdot, y, \cdot)$ for the case where the random strings for both A and B are chosen uniformly at random.

PRIVATE INFORMATION RETRIEVAL. Informally, a private information retrieval (PIR) scheme is an interactive protocol between two parties, a database $\mathcal{D}$ and a user $\mathcal{U}$. The database holds a *data string* $x \in \{0,1\}^n$, and the user holds an *index* $i \in [n]$. In its one-round version, the protocol consists of (a) a query sent from the user to the database (generated by an efficient randomized query algorithm, taking as an input the index i and a random string $r_{\mathcal{U}}$); (b) an answer sent by the database (generated by an efficient deterministic (without loss of generality) answer algorithm, taking as an input the query sent by the user and the database x); and (c) an efficient reconstruction function applied by the user (taking as an input the index i, the random string $r_{\mathcal{U}}$, and the answer sent by the database). At the end of the execution of the protocol, the following two properties must hold: (1) after applying the reconstruction function, the user obtains the i-th data bit x_i; and (2) the distributions on the query sent to the database are computationally indistinguishable for any two indices i, i'. (That is, a computationally bounded database does not receive any information about the index of the user). We now give a formal definition of a PIR scheme.

Definition 1 (Private Information Retrieval Scheme.) Let $(\mathcal{D}, \mathcal{U})$ be an interactive protocol, and let $\mathcal{R}$ be a polynomial time algorithm[2]. We say that $(\mathcal{D}, \mathcal{U}, \mathcal{R})$ is a *private information retrieval (PIR) scheme* if:

1. (*Correctness.*) For each $n \in \mathbb{N}$, each $i \in \{1, \ldots, n\}$, each $x \in \{0,1\}^n$, where $x = x_1 \circ \cdots \circ x_n$, and $x_l \in \{0,1\}$ for $l = 1, \ldots, n$, and for all constants c, and all sufficiently large k,

$$\text{Prob}[\,(r_{\mathcal{D}}, r_{\mathcal{U}}, t) \leftarrow t_{\mathcal{D},\mathcal{U}}(\,(1^k, x), \cdot, (1^k, n, i), \cdot) : \mathcal{R}(1^k, n, i, r_{\mathcal{U}}, t) = x_i\,] \geq 1 - k^{-c}.$$

[2] For clarity, we chose to include the reconstruction function $\mathcal{R}$ as an explicit part of the PIR definition. We note however that replacing $\mathcal{R}$ by $\mathcal{U}$ in the correctness requirement yields an equivalent definition (where the reconstruction function is an implicit part of $\mathcal{U}$, who executes it to produce an output).

2. (*User Privacy.*) For each $n \in \mathbb{N}$, each $i, j \in \{1, \ldots, n\}$, each $x \in \{0,1\}^n$, where $x = x_1 \circ \cdots \circ x_n$, and $x_l \in \{0,1\}$ for $l = 1, \ldots, n$, for each polynomial time $\mathcal{D}'$, for all constants c, and all sufficiently large k, it holds that $|p_i - p_j| \leq k^{-c}$, where

$$\begin{aligned} p_i &= \mathrm{Prob}[\,(r_{\mathcal{D}'}, r_{\mathcal{U}}, t) \leftarrow t_{\mathcal{D}',\mathcal{U}}(\,(1^k, x), \cdot, (1^k, n, i), \cdot) : \mathcal{D}'(1^k, x, r_{\mathcal{D}'}, t) = 1\,] \\ p_j &= \mathrm{Prob}[\,(r_{\mathcal{D}'}, r_{\mathcal{U}}, t) \leftarrow t_{\mathcal{D}',\mathcal{U}}(\,(1^k, x), \cdot, (1^k, n, j), \cdot) : \mathcal{D}'(1^k, x, r_{\mathcal{D}'}, t) = 1\,]. \end{aligned}$$

We say that $(\mathcal{D}, \mathcal{U}, \mathcal{R})$ is an *honest-database PIR scheme* if it is a PIR scheme in which the user-privacy requirement is relaxed to hold only for $\mathcal{D}'$ that follow the protocol execution as $\mathcal{D}$.

For sake of generality, the above definition does not pose any restriction on the number of rounds of protocol $(\mathcal{D}, \mathcal{U})$; however, we remark that the most studied case in the literature is that of one-round protocols (as discussed above).

SYMMETRICALLY PRIVATE INFORMATION RETRIEVAL. Informally, a symmetrically private information retrieval (SPIR) scheme is a PIR scheme satisfying an additional privacy property: data privacy. Namely, for each execution, there exists an index i, such that the distributions on the user's view are computationally indistinguishable for any two databases x, y such that $x_i = y_i$. (That is, a computationally bounded user does not receive information about more than a single bit of the data). We now give a formal definition of a SPIR scheme.

Definition 2 (Symmetrically Private Information Retrieval Scheme)
Let $(\mathcal{D}, \mathcal{U}, \mathcal{R})$ be a PIR scheme. We say that $(\mathcal{D}, \mathcal{U}, \mathcal{R})$ is a *symmetrically private information retrieval (SPIR) scheme* if in addition it holds that

3. (*Data Privacy.*) For each $n \in \mathbb{N}$, for each polynomial time $\mathcal{U}'$, each $i' \in \{1, \ldots, n\}$, and each random string $r_{\mathcal{U}'}$, there exists an $i \in \{1, \ldots, n\}$, such that for each $x, y \in \{0,1\}^n$ where $x = x_1 \circ \cdots \circ x_n$ and $y = y_1 \circ \cdots \circ y_n$, $x_l, y_l \in \{0,1\}$ for $l = 1, \ldots, n$, and such that $x_i = y_i$, for all constants c and all sufficiently large k, it holds that $|p_x - p_y| \leq k^{-c}$, where

$$\begin{aligned} p_x &= \mathrm{Prob}[\,(r_{\mathcal{D}}, t) \leftarrow t_{\mathcal{D},\mathcal{U}'}((1^k, x), \cdot, (1^k, n, i'), r_{\mathcal{U}'}) : \mathcal{U}'(1^k, n, i', r_{\mathcal{U}'}, t) = 1\,] \\ p_y &= \mathrm{Prob}[\,(r_{\mathcal{D}}, t) \leftarrow t_{\mathcal{D},\mathcal{U}'}((1^k, y), \cdot, (1^k, n, i'), r_{\mathcal{U}'}) : \mathcal{U}'(1^k, n, i', r_{\mathcal{U}'}, t) = 1\,]. \end{aligned}$$

OBLIVIOUS TRANSFER. Informally, a $\binom{2}{1}$-Oblivious Transfer ($\binom{2}{1}$-OT) is an interactive protocol between Alice, holding two bits b_0, b_1, and Bob, holding a selection bit c. At the end of the protocol, Bob should obtain the bit b_c, but no information about $b_{\bar{c}}$, whereas Alice should obtain no information about c. (By "obtaining no information" we mean that the two possible views are indistinguishable.) The extension to $\binom{n}{1}$-OT is immediate. A formal definition follows.

Definition 3 ($\binom{2}{1}$-Oblivious Transfer)

Let (Alice, Bob) be an interactive protocol. We say that (Alice, Bob) is a $\binom{2}{1}$*-Oblivious Transfer* ($\binom{2}{1}$*-OT*) protocol with security parameter k if it holds that:

1. *(Correctness).* For all $b_0, b_1, c \in \{0,1\}$, all constants d, and all sufficiently large k,

$$\mathrm{Prob}[\,(r_A, r_B, t) \leftarrow t_{\text{Alice,Bob}}((1^k, b_0, b_1), \cdot, (1^k, c), \cdot) : \mathrm{Bob}(1^k, c, r_B, t) = b_c\,] \geq 1 - k^{-d}.$$

2. *(Privacy against Alice).* For all probabilistic polynomial time Alice$'$, all $b_0, b_1 \in \{0,1\}$, all constants d, and all sufficiently large k,

$$\begin{aligned}\mathrm{Prob}\ \ [\ \ & c \leftarrow \{0,1\}; (r_{A'}, r_B, t) \leftarrow t_{\text{Alice}',\text{Bob}}((1^k, b_0, b_1), \cdot, (1^k, c), \cdot) : \\ & \mathrm{Alice}'(1^k, b_0, b_1, r_{A'}, t) = c\,] \leq 1/2 + k^{-d}.\end{aligned}$$

3. *(Privacy against Bob).* For all probabilistic polynomial time Bob$'$, all $c' \in \{0,1\}$, and all random strings $r_{B'}$, there exists $c \in \{0,1\}$ such that for all constants d, and all sufficiently large k,

$$\begin{aligned}\mathrm{Prob}\ \ [\ \ & (b_0, b_1) \leftarrow \{0,1\}^2; (r_A, t) \leftarrow t_{\text{Alice,Bob}'}((1^k, b_0, b_1), \cdot, (1^k, c'), r_{B'}) : \\ & \mathrm{Bob}'(1^k, c', r_{B'}) = b_{\bar{c}}\,] \leq 1/2 + k^{-d}.\end{aligned}$$

We say that (Alice, Bob) is an *honest-Bob-*$\binom{2}{1}$*-OT* protocol if it is a $\binom{2}{1}$-OT protocol in which privacy against Bob is relaxed to hold only when Bob is honest (but curious). That is, condition (3) in Definition 3 is relaxed to

3'. *(Privacy against honest-but-curious-Bob).* For all probabilistic polynomial time CuriousB, for all constants d, and all sufficiently large k,

$$\begin{aligned}\mathrm{Prob}\ \ [\ \ & (b_0, b_1) \leftarrow \{0,1\}^2; (r_A, r_B, t) \leftarrow t_{\text{Alice,Bob}}((1^k, b_0, b_1), \cdot, (1^k, c), \cdot) : \\ & \mathrm{CuriousB}(1^k, c, r_B, t) = b_{\bar{c}}\,] \leq 1/2 + k^{-d}.\end{aligned}$$

We say that (Alice, Bob) is an *honest-parties-*$\binom{2}{1}$*-OT* protocol if it is a $\binom{2}{1}$-OT protocol where privacy requirements are relaxed to hold only when both Alice and Bob are honest-but-curious; that is, (Alice, Bob) should satisfy correctness, privacy against honest-but-curious Bob (as defined above), and privacy against honest-but-curious Alice (which is similarly defined).

We remark that the definitions of $\binom{2}{1}$-OT and its honest-but-curious versions are extended in the obvious way to the case of $\binom{n}{1}$-OT, for any $n \geq 3$.

COMMUNICATION COMPLEXITY. Let $(\mathcal{D},\mathcal{U},\mathcal{R})$ be a PIR scheme. We define the *communication complexity* of $(\mathcal{D},\mathcal{U},\mathcal{R})$ as the maximal length $c(n)$ of a transcript returned by a possible execution of $(\mathcal{D},\mathcal{U},\mathcal{R})$ where n is the size of $\mathcal{D}$'s input (i.e. the length of the database). We define the *database communication complexity* as the maximal length $c_{\mathcal{D}}(n)$ of the communication sent by $\mathcal{D}$ in any execution of $(\mathcal{D},\mathcal{U},\mathcal{R})$, and similarly the *user communication complexity* $c_{\mathcal{U}}(n)$. That is, $c(n) = c_{\mathcal{D}}(n) + c_{\mathcal{U}}(n)$. The communication complexity of a SPIR scheme and of an $\binom{n}{1}$-OT scheme are similarly defined.

SPIR VS. $\binom{n}{1}$-OT. It can be easily verified that $\binom{n}{1}$-OT is equivalent to SPIR with a database of length n. The reason we need two concepts (and the reason we formulated the definitions in two different, though equivalent, ways), is the different motivations for using these primitives (and the way they were historically defined). In particular, we note that when constructing a SPIR protocol, the communication complexity is a crucial parameter.

3 PIR Implies honest-Bob-$\binom{2}{1}$-OT

In this section we construct an honest-Bob-$\binom{2}{1}$-OT protocol from any PIR scheme with database communication complexity $c_{\mathcal{D}}(k) < k$ (and arbitrary user communication complexity $c_{\mathcal{U}}(k)$), for database of length k.[3]

THE PROTOCOL DESCRIPTION. Let $\mathcal{P} = (\mathcal{D},\mathcal{U},\mathcal{R})$ be a PIR scheme with database communication $c_{\mathcal{D}}(k) < k$. Our $\binom{2}{1}$-OT protocol consists of simultaneously invoking polynomially many[4] independent executions of $\mathcal{P}$ with a random data string for $\mathcal{D}$ (ran by Alice) and random indices for $\mathcal{U}$ (ran by Bob). In addition, Bob sends to Alice two sequences of indices (one consists of the indices retrieved in the PIR invocations, and one a sequence of random indices), and in response Alice sends to Bob her two secret bits appropriately masked, so that Bob can reconstruct only one of them. A formal description of protocol (Alice, Bob) is in Figure 1. We note that some related techniques to those in our construction have appeared in [5]; however, we remark that the protocol of [5] cannot be used in our case, mainly because of the differences in the models. We next prove that (Alice, Bob) is a honest-Bob-$\binom{2}{1}$-OT protocol.

CORRECTNESS. In order to prove the correctness of (Alice, Bob), we need to show that Bob outputs b_c with probability at least $1 - k^{-\omega(1)}$. First, notice that if Bob is able to correctly reconstruct all bits $x^j(i^j)$ for $j = 1,\ldots,m$, after the m executions of the PIR protocol in step 1, then he is able to compute the right value for b_c in step 5. Next, from the

[3] In this section and the next we denote the database length by k, since the way it will be used will be for a database whose length depends (polynomially) on the security parameter. This is to avoid confusion with the length of the actual database n in the last section, where we construct SPIR using this $\binom{2}{1}$-OT .

[4] The number of invocations, m, is a parameter whose value can be set based on the communication complexity of $\mathcal{P}$ and the target (negligible) probability of error in OT, but will always be polynomial in k as will become clear below.

correctness of $\mathcal{P} = (\mathcal{D}, \mathcal{U}, \mathcal{R})$, Bob, who is playing as $\mathcal{U}$, is able to reconstruct all bits $x^j(i^j)$ with probability at least $(1 - k^{-\omega(1)})^m$ since the m executions of $(\mathcal{D}, \mathcal{U})$ are all independent. This probability is then at least $1 - k^{-\omega(1)}$ since m is polynomial in k.

Privacy against Alice. In order to prove that (Alice, Bob) satisfies the property of privacy against Alice, we need to show that for any probabilistic polynomial time algorithm Alice′, the probability that Alice′, at the end of the protocol, is able to compute the bit c input to Bob is at most $1/2 + k^{-\omega(1)}$ (where probability is taken over the uniform distribution of c and the random strings of Alice′ and Bob). Informally, this follows from the user's privacy in the PIR subprotocol $\mathcal{P}$, which guarantees that in each invocation Alice gets no information about the index used by Bob, and thus cannot tell between the sequence of real indices used, and the sequence of random indices (since both these sequences are distributed uniformly). A more formal argument follows. Assume for the sake of contradiction that the property is not true; namely, there exists a probabilistic polynomial time algorithm Alice′, which, after running protocol (Alice′, Bob), is able to compute c with probability at least $1/2 + k^{-d}$, for some constant d and infinitely many k. In step 3, Bob sends two m-tuples (I_0, I_1) of indices to Alice′, such that I_c is the tuple of indices used by Bob in the PIR invocations of step 1, and $I_{\bar{c}}$ is a tuple containing random indices. Therefore, Alice′ is able to guess with probability at least $1/2 + k^{-d}$ which one of I_0, I_1 is the tuple of retrieved indices. This implies, by a hybrid argument, that for some position $j \in \{1, \ldots, m\}$, Alice′ can guess with probability at least $1/2 + k^{-d}/m$ whether in the j-th PIR invocation the index used was i_0^j or i_1^j. Since all PIR invocations are independent (implying that the indices in different positions within I_0 and I_1 are independent), it is straightforward to use Alice′ to construct a $\mathcal{D}'$ which distinguishes in a single PIR execution between the index used by the user and a random index, with probability at least $1/2 + k^{-d}/m$. Since m is polynomial, this is a non-negligible advantage, and thus contradicts the user privacy of $\mathcal{P}$.

Privacy against honest-but-curious Bob. In order to prove that the pair (Alice, Bob) satisfies the property of privacy against a honest-but-curious Bob, we need to show that the probability that Bob, after behaving honestly in the protocol, is able to compute the bit $b_{\bar{c}}$ is at most $1/2 + k^{-\omega(1)}$ (where probability is taken over the uniform distribution of b_0, b_1, and the random strings of Alice and Bob). In order to prove this property for an appropriate polynomial number m of invocations of $(\mathcal{D}, \mathcal{U})$ in step 1, we start by considering a single invocation. In the following lemma we consider the probability p that a malicious user $\mathcal{U}'$, after invoking $(\mathcal{D}, \mathcal{U}')$ where $\mathcal{D}$ uses a uniformly chosen database, fails in reconstructing a bit in a *random* location j in the database. Note that j is not known to $\mathcal{U}'$ when running $(\mathcal{D}, \mathcal{U}')$.[5] We also note that no further requirements about $\mathcal{U}'$ or its computational power are necessary. In the following we show that if the database communication complexity is

[5]Indeed, if $\mathcal{U}'$ had known which location j he would have to reconstruct, he could run the honest user algorithm $\mathcal{U}$ with input j, and could reconstruct the correct bit with high probability using the reconstruction function $\mathcal{R}$.

Honest-Bob-$\binom{2}{1}$-OT

Alice's inputs: 1^k (where k is a security parameter) and $b_0, b_1 \in \{0,1\}$.
Bob's inputs: 1^k and $c \in \{0,1\}$.
Additional (common) inputs: a parameter m polynomial in k, and a PIR protocol $(\mathcal{D}, \mathcal{U}, \mathcal{R})$.

Instructions for Alice and Bob:

1. For every $j \in \{1, \ldots, m\}$ do:
   - Alice uniformly chooses a data string $x^j \in \{0,1\}^k$ (where x^j can be written as $x^j(1) \circ \cdots \circ x^j(k)$, for $x^j(i) \in \{0,1\}$).
   - Bob uniformly chooses an index $i^j \in [k]$
   - Alice and Bob invoke the PIR protocol $(\mathcal{D}, \mathcal{U}, \mathcal{R})$ where Alice plays the role of $\mathcal{D}$ on input $(1^k, x^j)$ and Bob plays the role of $\mathcal{U}$ on input $(1^k, k, i^j)$. (That is, Alice and Bob execute $(\mathcal{D}, \mathcal{U})$ on the above inputs, and then Bob applies the reconstruction function $\mathcal{R}$ to obtain the bit $x^j(i^j)$).
2. Bob sets $(i_c^1, \ldots, i_c^m) \stackrel{\text{def}}{=} (i^1, \ldots, i^m)$ `(*the indices retrieved*)`

   and uniformly chooses $(i_{\bar{c}}^1, \ldots, i_{\bar{c}}^m)$ from $[k]^m$. `(*random indices*)`
3. Bob sends to Alice $(i_0^1, \ldots, i_0^m)$ and $(i_1^1, \ldots, i_1^m)$.
4. Alice sets $z_0 \stackrel{\text{def}}{=} b_0 \oplus x^1(i_0^1) \oplus \ldots \oplus x^m(i_0^m)$, and $z_1 \stackrel{\text{def}}{=} b_1 \oplus x^1(i_1^1) \oplus \ldots \oplus x^m(i_1^m)$

   and sends z_0, z_1 to Bob;
5. Bob computes $b_c = z_c \oplus x^1(i^1) \oplus \ldots \oplus x^m(i^m)$ and **outputs:** b_c.

Figure 1: A protocol (Alice,Bob) for honest-Bob-$\binom{2}{1}$-OT , using a PIR protocol $\mathcal{P} = (\mathcal{D}, \mathcal{U}, \mathcal{R})$ with $c_{\mathcal{D}}(k) < k$ database communication complexity.

less than the length of the data, this failure probability is non-negligible. This is shown by first bounding the binary entropy of the failure probability.

Lemma 1 *Let $\mathcal{P} = (\mathcal{D}, \mathcal{U}, \mathcal{R})$ be a PIR scheme with database communication complexity $c_{\mathcal{D}}(k)$. For every interactive Turing machine $\mathcal{U}'$, every reconstruction algorithm $\mathcal{R}'$, every $r_{\mathcal{U}'}$, and every k, let*

$$p \stackrel{\text{def}}{=} \text{Prob}\left[\begin{array}{l} x = x_1 \circ \cdots \circ x_k \leftarrow \{0,1\}^k;\ (r_{\mathcal{D}}, t) \leftarrow t_{\mathcal{D},\mathcal{U}'}((1^k, x), \cdot, 1^k, r_{\mathcal{U}'}); \\ \qquad\qquad j \leftarrow [k] : \mathcal{R}'(1^k, r_{\mathcal{U}'}, t, j) \neq x_j \end{array}\right]$$

Then it holds that $H(p) \geq \frac{k - c_{\mathcal{D}}(k)}{k}$, where $H(p)$ is the binary entropy function $H(p) \stackrel{\text{def}}{=} p \log(1/p) + (1-p)\log(1/(1-p))$.

Proof. We need to prove that, for every $\mathcal{U}'$ and $\mathcal{R}'$, after running $(\mathcal{D}, \mathcal{U}')$ with a uniform data string for $\mathcal{D}$, the probability that $\mathcal{R}'$ fails in reconstructing a data bit in a uniformly chosen location j, has binary entropy which is bounded below by $\frac{k - c_{\mathcal{D}}(k)}{k}$. This is proved using standard information theory arguments (e.g., similar arguments have been used in [3]). For background and terminology used in the proof below, see for example [9].

Let X be the random variable ranging over the data strings (where X_j corresponds to the j-th bit), and A be the random variable ranging over the database answers. Thus, the length of A is at most $c_{\mathcal{D}}(k)$, implying that $H(A) \leq c_{\mathcal{D}}(k)$ (where H is the entropy function for random variables). Let $\hat{X} \in \{0,1\}^k$ denote the user's reconstruction of the data string X, namely (following the notation in the lemma), $\hat{X}_j = \mathcal{R}'(1^k, r_{\mathcal{U}'}, t, j)$ for $j \in [k]$. Let $p_j \stackrel{\text{def}}{=} \text{Prob}[\hat{X}_j \neq X_j]$ be the probability of failure in reconstructing the j-th bit. The probability of failure in reconstructing a *random* bit-location is therefore $p = (1/k) \cdot \sum_{j=1}^{k} p_j$. By Fano's inequality (see [9]), we have that $H(p_j) \geq H(X_j|A)$, for all $j = 1, \ldots, k$, where $H(p_j)$ refers to the binary entropy function, and $H(X_j|A)$ is the entropy of X_j given A. By the chain rule for entropy,

$$H(X|A) = \sum_{j=1}^{k} H(X_j|A, X_{j-1}, \ldots X_1) \leq \sum_{j=1}^{k} H(X_j|A)$$

On the other hand,

$$H(X|A) = H(X) - H(A) + H(A|X) = k - H(A) \geq k - c_{\mathcal{D}}(k),$$

where the last equality follows since A is determined by X. Putting all the above together and using the concavity of the entropy function, we obtain that

$$H(p) = H(\frac{1}{k}\sum_{j=1}^{k} p_j) \geq \frac{1}{k}\sum_{j=1}^{k} H(p_j) \geq \frac{1}{k}\sum_{j=1}^{k} H(X_j|A) \geq \frac{H(X|A)}{k} \geq \frac{k - c_{\mathcal{D}}(k)}{k}$$

■

Remark 1 Note that Lemma 1 holds even when $c_{\mathcal{D}}(k)$ is defined as the *expected* database communication complexity (rather than the worst-case one). This is because the proof above holds for any $c_{\mathcal{D}}(k) \geq H(A)$, and indeed the expected length of A is bounded below by the entropy of A (according to the entropy bound on data compression [9]).

The relation between the failure probability p and its binary entropy is given by the following fact (the proof follows from the expression for the entropy function and is omitted).

Fact 1 *For every $\epsilon > 0$ there exists a constant $c > 0$ such that for every $0 \leq p < c$, $p\log(1/p) \leq H(p) \leq (1+\epsilon)p\log(1/p)$.*

The above fact allows us to translate the lower bound on $H(p)$ into a lower bound on p. For example, a loose manipulation of the fact yields that, for any $\delta > 0$ and small enough p, $p > H(p)^{1+\delta}$. More generally, if $H(p)$ is non-negligible then p is also non-negligible. For sake of concreteness, we state a corollary bounding the failure probability, using $\delta = 1$. This will be sufficient for our needs, although as explained tighter corollaries can be derived.

Corollary 1 *Let $\mathcal{P} = (\mathcal{D},\mathcal{U},\mathcal{R})$ be a PIR scheme with database communication complexity $c_{\mathcal{D}}(k)$. The there exists a constant $c > 0$ such that for every interactive Turing machine $\mathcal{U}'$, every reconstruction algorithm $\mathcal{R}'$, every $r_{\mathcal{U}'}$, and every k, letting p be as in Lemma 1, we have that either $p > c$, or $p \geq (1 - c_{\mathcal{D}}(k)/k)^2$.*

Thus, if the communication complexity $c_{\mathcal{D}}(k) < k$, the probability that the user fails to reconstruct a bit in a random location after a single execution is non-negligible. For example, if $c_{\mathcal{D}}(k) = k - 1$ this failure probability is at least $1/poly(k)$, and if $c_{\mathcal{D}}(k) \leq k/2$ the failure probability is constant.

Finally, recall that in our protocol Alice and Bob run m independent invocations of $(\mathcal{D},\mathcal{U})$, and (since Bob is honest-but-curious), $I_{\bar{c}} = (i^1_{\bar{c}}, \ldots, i^m_{\bar{c}})$ is a uniformly chosen m-tuple, independent of the random choices made in the PIR invocations. Moreover, Bob is able to reconstruct $b_{\bar{c}}$ if and only if he can reconstruct the exclusive-or of all values $x^1(i^1_{\bar{c}}) \oplus \cdots \oplus x^m(i^m_{\bar{c}})$, since he receives $z_{\bar{c}}$ from Alice in step 4. This, together with Corollary 1, yields that for an appropriately chosen polynomial number m, the failure probability is exponentially close to 1, namely Bob's probability of correctly reconstructing $b_{\bar{c}}$ is negligible. We conclude that our protocol maintains privacy against honest-but-curious Bob.

We have proved that the protocol of Figure 1 maintains correctness, privacy against Alice, and privacy against honest-but-curious Bob. We have therefore proved the following theorem.

Theorem 1 *If there exists a single database PIR scheme with database communication complexity $c_{\mathcal{D}}(k) < k$, where k is the length of the database, then there exists an honest-Bob-$\binom{2}{1}$-OT protocol with security parameter k.*

Similarly, it is easy to see that using a PIR scheme for which the data privacy requirement holds with respect to honest databases (rather than maliciously ones) in the protocol of Figure 1 yields an $\binom{2}{1}$-OT protocol for which both privacy requirement hold with respect to honest Alice and Bob.

Theorem 2 *If there exists a honest-database PIR scheme with database communication complexity $c_{\mathcal{D}}(k) < k$, where k is the length of the database, then there exists an honest-parties-$\binom{2}{1}$-OT protocol with security parameter k.*

The following remarks about the full strength of Theorem 1 follow from the proof above.
Round and Communication Complexity. Our protocol for honest-Bob-$\binom{2}{1}$-OT requires the same number of rounds as the underlying PIR protocol $\mathcal{P}$, and in particular if $\mathcal{P}$ has one round, so is the new protocol. This is so, since all the messages that need to be sent by Bob (in steps 1,3 of our protocol) can be computed in parallel and sent to Alice in a single message, and similarly all messages that need to be sent back by Alice (in steps 1,4) can be sent to Bob in a single message. We also note that our theorem holds even when we consider *expected* communication complexity (rather than maximal).
Computational Power of the Parties. Our transformation from PIR to honest-Bob-$\binom{2}{1}$-OT preserves the computational power of the parties; namely, if $\mathcal{D}$ (resp., $\mathcal{U}$) runs in polynomial time, then so does Alice (resp., Bob). In terms of privacy, our result is stronger than stated in Theorem 1; namely, the privacy against the honest-but-curious Bob is information-theoretic (to see this, observe that in the proof of this property we never make any assumption on the computational power of Bob, but rather rely on Lemma 1 which is information-theoretic). On the other hand, the privacy against Alice requires the same assumptions as on the computational power of $\mathcal{D}$ in the PIR protocol $(\mathcal{D},\mathcal{U})$; however, notice that Alice *must* be computationally bounded, since there exists no single database PIR protocol with communication complexity smaller than the size of the database and private against a computationally unbounded database [8].
Our Reduction. We note that our construction is a *black-box* reduction in the following sense: the $\binom{2}{1}$-OT uses the underlying PIR protocol as a subroutine with the only guarantee that the total number of bits that user gets regarding the database is strictly less then the total size of the database (i.e., without relying on any specific features of the implementation, and without making any additional assumptions about the implementation.) Thus any idealized implementation of this primitive (as a black-box) will also work for our purposes. As a consequence, our reduction is also "code-to-code". That is, any *implementation* of non-trivial Single-Database PIR protocol will also give an implementation of OT. In this aspect, our reduction is similar to [4].

4 PIR Implies $\binom{2}{1}$-OT (even for dishonest parties)

In this section, we transform the protocol given in Figure 1 into a protocol that is resilient against arbitrary (possibly dishonest) parties. That is, we prove the following analogue of Theorem 1.

Theorem 3 *If there exists a single database PIR scheme with database communication complexity $c_{\mathcal{D}}(k) < k$, where k is the length of the database, then there exists an $\binom{2}{1}$-OT protocol* *with security parameter k. Moreover, if the original PIR scheme requires a constant number of rounds then so does the resulting $\binom{2}{1}$-OT protocol.*

Proof. Let $\mathcal{P}$ be a PIR scheme with database communication $c_{\mathcal{D}}(k) < k$. Theorem 1 guarantees an implementation of $\binom{2}{1}$-OT for honest-but-curious Bob. Such an implementation may be transformed into one for dishonest parties, using (by now standard) techniques originating in [18, 19], based on commitment schemes and zero-knowledge proofs for NP-complete languages. The resulting reduction, however, would return a protocol for $\binom{2}{1}$-OT having a number of rounds polynomial in k even if the original PIR scheme has a constant number of rounds. Below we sketch a more direct reduction, combining ideas in [19] with techniques for witness-indistinguishability protocols from [14], which yields a constant round $\binom{2}{1}$-OT whenever $\mathcal{P}$ is a constant round PIR.

Let us denote by (Alice, Bob) the $\binom{2}{1}$-OT scheme obtained applying Theorem 1 to $\mathcal{P}$. In order to achieve privacy against a possibly dishonest Bob, it is enough to design the scheme so that the following two properties are satisfied: (1) the two m-tuples of indices $(i_0^1, \ldots, i_0^m)$ and $(i_1^1, \ldots, i_1^m)$ are uniformly and independently distributed over $[n]^m$; (2) Bob's messages during the execution of the PIR subprotocols are computed according to the specified program, and using randomness that is independently distributed from the above two m-tuple of indices. In order to achieve the first property, the two m-tuples are computed using a flipping coin subprotocol at the beginning of protocol (Alice,Bob). In order to achieve the second property, at the beginning of the protocol Bob commits to the randomness to be later used while running the PIR subprotocol. Specifically, the protocol (Alice,Bob) is modified as follows.

At the beginning of protocol (Alice, Bob):

1. Bob commits to a sufficiently random string R and to randomly chosen indices $(l_0^1, \ldots, l_0^m)$ and $(l_1^1, \ldots, l_1^m)$ by sending three commitment keys com_R, com_0, com_1;
2. Alice sends random indices $(h_0^1, \ldots, h_0^m)$ and $(h_1^1, \ldots, h_1^m)$;
3. Bob sets $i_d^j = (h_d^j + l_d^j \bmod n) + 1$, for $j = 1, \ldots, m$ and $d = 0, 1$;

When required to use indices $(i^1, \ldots, i^m)$ in step 1 of (Alice, Bob), for each message he sends:

4. Bob proves that the message has been correctly computed according to the PIR subprotocol, using the string R committed in step 1 above as random tape, and using as a tuple of indices one of the two m-tuples committed in step 1 above. This can be written as an NP statement and can be efficiently reduced to a membership statement T for an NP complete language. Bob proves T to Alice by using a witness-indistinguishable proof system.

When required to send indices $(i_d^1, \ldots, i_d^m)$, for $d = 0, 1$, in step 3 of (Alice, Bob):

5. Bob proves that the two tuples he is sending have been correctly computed in the following sense: one is the same used in the PIR subprotocols and one is the one

out of the two committed in step 1 above not used in the PIR subprotocols. This can be written as an NP statement and can be efficiently reduced to a membership statement T for an NP complete language. Bob proves T to Alice by using a witness-indistinguishable proof system.

We note that the parallel execution of an atomic zero-knowledge proof system for an NP-complete language as the one in [18] is known to be witness-indistinguishable from results in [14] and can be implemented using only 3 rounds of communication, and therefore can be used in steps 4 and 5 above.

Now, let us briefly show that the modified protocol (Alice, Bob) is a $\binom{2}{1}$-OT protocol. First of all, observe that the described modification does not affect the property of correctness, which therefore continues to hold. Then observe that the fact that the privacy against Alice continues to hold follows from the witness-indistinguishability of the proof system used, and the privacy against a possibly dishonest Bob follows from the soundness of the proof system used. Moreover, the overall number of rounds of the modified protocol (Alice, Bob) is constant and no additional complexity assumption is required, since commitment schemes and 3-round witness-indistinguishable proof systems for NP complete languages can be implemented using any one-way function [20, 30] and one-way functions, in turn, can be obtained by any low-communication PIR protocol [3]. ■

We remark that in the case $c(k) < k/2$ the above transformation can be made more efficient (by a polynomial factor) using a direct derivation of commitment schemes from low communication PIR, provided in [3]. Finally, using Theorem 2 and the same techniques as above, Theorem 3 can be strengthened to transform even an honest-database PIR into a $\binom{2}{1}$-OT protocol; that is:

Theorem 4 *If there exists a single database honest-database PIR scheme with database communication complexity $c_{\mathcal{D}}(k) < k$, where k is the length of the database, then there exists an $\binom{2}{1}$-OT protocol with security parameter k.*

5 PIR Implies SPIR

We are now ready to complete the proof of the following theorem.

Theorem 5 *If there exists a single database PIR scheme with communication complexity $c(n) < n$, where n is the length of the database, then there exists a single database SPIR scheme with security parameter k and communication complexity $c(n) + q(k)$ for some polynomial q.*

Proof. First, by the result of Naor and Pinkas [31], we know that given a family of pseudo-random functions, a $\binom{2}{1}$-OT primitive, and a single database PIR with communication complexity $c(n)$, there exists a single database SPIR protocol which uses $\log n$ invocations of

$\binom{2}{1}$-OT , and one call to PIR protocol with communication complexity $c(n)$ where n is the length of the data string and k is the security parameter. Next, since PIR implies one-way functions (first proved in [3] and also directly follows from the results in the previous section), PIR also implies pseudo-random functions [17, 20]. Finally, by our result in the previous section, PIR implies $\binom{2}{1}$-OT (where the communication complexity is some polynomial $poly'$ in the security parameter). Thus, we get that PIR implies SPIR with communication complexity $c'(n)$, satisfying $c'(n) = c(n) + poly(k) \log n = c(n) + poly'(k)$ where $poly, poly'$ are polynomials, k is a security parameter, and n is the length of the database. ∎

Acknowledgments We thank Amos Beimel, Yevgeniy Dodis, Oded Goldreich, Yuval Ishai and Silvio Micali for useful comments.

References

[1] A. Ambainis. Upper Bound on the Communication Complexity of Private Information Retrieval. In *Proc. of 24th ICALP*, 1997.

[2] G. Brassard, C. Crepeau and J.-M. Robert All-or-Nothing Disclosure of Secrets In *Crypto '86* Springer-Verlag, 1987, pp. 234-238.

[3] A. Beimel, Y. Ishai, E. Kushilevitz, and T. Malkin. One-Way Functions are Essential for Single-Server Private Information Retrieval. In *Proc. of 31st STOC*, 1999.

[4] A. Beimel, T. Malkin, S. Micali. The All-or-Nothing Nature of Two-Party Secure Computation. In *Proc. of CRYPTO 99*, 1999.

[5] C. Cachin, C. Crepeau, and S. Marcil. Oblivious Transfer with a Memory Bounded Receiver. In *Proc. of 39th FOCS*, 1998.

[6] C. Cachin, S. Micali, and M. Stadler. Computationally Private Information Retrieval with Polylogarithmic Communication. In *Proc. of EUROCRYPT 99*, 1999.

[7] B. Chor and N. Gilboa. Computationally Private Information Retrieval. In *Proc. of 29th STOC*, 1997.

[8] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private Information Retrieval. In *Proc. of 36th FOCS*, 1995.

[9] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley & Sons, 1991.

[10] C. Crépeau. Equivalence Between Two Flavors of Oblivious Transfers. In *Proc. of CRYPTO '87*, 1987.

[11] A. De Santis and P. Persiano, Zero-Knowledge Proofs of Knowledge without Interaction, In *Proc. of 33rd FOCS*, 1992.

[12] G. Di Crescenzo, Y. Ishai, and R. Ostrovsky. Universal Service-Providers for Database Private Information Retrieval. In *Proc. of 17th PODC*, 1998.

[13] S. Even, O. Goldreich, and A. Lempel. A Randomized Protocol for Signing Contracts. *Comm. of ACM*, 28:637–647, 1985.

[14] U. Feige and A. Shamir, Witness Indistinguishable and Witness Hiding Protocols. In *Proc. of 23rd STOC*, 1990.

[15] Y. Gertner, S. Goldwasser, and T. Malkin. A Random Server Model for Private Information Retrieval. In *Proc. of 2nd RANDOM*, 1998.

[16] Y. Gertner, Y. Ishai, E. Kushilevitz, and T. Malkin. Protecting Data Privacy in Private Information Retrieval Schemes. In *Proc. of 30th STOC*, 1998.

[17] O. Goldreich, S. Goldwasser, and S. Micali. How to Construct Random Functions. In *Journal of the ACM*, vol. 38, 1991, pages 792–807, 1986.

[18] O. Goldreich, S. Micali, and A. Wigderson. Proofs that Yield Nothing but Their Validity, and a Methodology of Cryptographic Protocol Design. In *Journal of the ACM*, vol. 38, 1991, pages 691–729.

[19] O. Goldreich, S. Micali, and A. Wigderson. How to Play Any Mental Game. In *Proc. of 19th STOC*, 1987.

[20] J. Hastad, R. Impagliazzo, L. Levin, and M. Luby. A Pseudorandom Generator From Any One-Way Function. *SIAM J. on Computing*, 1999, vol. 28, n. 4.

[21] R. Impagliazzo and M. Luby, "One-way Functions are Essential for Complexity-Based Cryptography" In *Proc. of FOCS 89.*

[22] R. Impagliazzo and S. Rudich. Limits on the Provable Consequences of One-Way Permutations. In *Proc. of 21st STOC*, 1989.

[23] Y. Ishai and E. Kushilevitz. Improved Upper Bounds on Information-Theoretic Private Information Retrieval. In *Proc. of 31st STOC*, 1999.

[24] J. Kilian. Founding Cryptography on Oblivious Transfer. In *Proc. of 20th STOC*, 1988.

[25] J. Kilian, E. Kushilevitz, S. Micali, and R. Ostrovsky. Reducibility and Completness In Private Computations. In *SIAM J. on Computing*, includes [26].

[26] E. Kushilevitz, S. Micali, and R. Ostrovsky. Reducibility and Completness in Multi-party private computations. In *Proc. of 35th FOCS 94*, 1994.

[27] E. Kushilevitz and R. Ostrovsky. Replication is Not Needed: Single-Database Computationally Private Information Retrieval. In *Proc. of 38th FOCS*, 1997.

[28] E. Kushilevitz and R. Ostrovsky. One-way Trapdoor Permutations are Sufficient for Single-database Computationally-Private Information Retrieval. In *Proc. of EUROCRYPT '00*, 2000.

[29] E. Mann. Private Access to Distributed Information. Master's thesis, Technion - Israel Institute of Technology, Haifa, 1998.

[30] M. Naor. Bit Commitment Using Pseudorandom Generators. *Journal of Cryptology*, 4:151–158, 1991.

[31] M. Naor and B. Pinkas. Oblivious Transfer and Polynomial Evaluation. In *Proc. of 31st STOC*, 1999.

[32] M. Naor and M. Yung. Universal One-way Hash Functions and their Cryptographic Applications. In *Proc. of 21st STOC*, 1989.

[33] R. Ostrovsky and V. Shoup. Private Information Storage. In *Proc. of 29th STOC*, 1997.

[34] R. Ostrovsky and A. Wigderson One-Way Functions are Essential for Non-Trivial Zero-Knowledge. In *Proceedings of the Second Israel Symposium on Theory of Computing and Systems* (ISTCS-93) pp. 1-10., IEEE 1993.

[35] M. O. Rabin. How to Exchange Secrets by Oblivious Transfer. Technical Report TR-81, Harvard University, 1981.

[36] J. Rompel. One-way Functions are Necessary and Sufficient for Secure Signatures. In *Proc. of 22nd STOC*, 1990.

[37] J. P. Stern. A New and Efficient All-or-Nothing Disclosure of Secrets Protocol. In *Proc. of ASIACRYPT '98*, 1998.