Electronic Colloquium on Computational Complexity, Comment 1 on Report No. 87 (2003)

A TIGHT LOWER BOUND FOR RESTRICTED PIR PROTOCOLS

Richard Beigel, Lance Fortnow,
and William Gasarch

Abstract. We show that any 1-round 2-server Private Information Retrieval Protocol where the answers are 1-bit long must ask questions that are at least $n-2$ bits long, which is nearly equal to the known $n-1$ upper bound. This improves upon the approximately $0.25n$ lower bound of Kerenidis and de Wolf while avoiding their use of quantum techniques.

Keywords. Private information retrieval, lower bounds, privacy
Subject classification. 68Q17,

1. Introduction

Following prior papers on Private Information Retrieval Protocols (Ambainis 1997; Cachin *et al.* 1999; Chor *et al.* 1998; Gasarch 2004; Kushilevitz & Ostrovsky 1997) we model a database as an n-bit string $x = x_1 \cdots x_n$. Suppose that Alice wants to know x_i but does not want the database to obtain any information about i. We do not impose any computational limits on the database, though some researchers have considered such limits (Cachin *et al.* 1999; Chor & Gilboa 1997; Kushilevitz & Ostrovsky 1997). If there is only one copy of the database then the only way to ensure privacy is to request the entire string x, which is n bits long. If there are $k \geq 2$ copies of the database that do not communicate with each other then the number of bits can be reduced. We refer to a copy of the database as a *server*.

Many upper bounds have been obtained. These include

1. If there are two servers then $O(n^{1/3})$ bits of communication suffice (Chor *et al.* 1998).

2. If there are k servers then $O(n^{1/(2k-1)})$ bits of communication suffice (Ambainis 1997; Beimel *et al.* 2002).

3. If there are k servers then $n^{O(\log\log k/k \log k)}$ bits of communication suffice (Beimel *et al.* 2002).

ISSN 1433-8092

Lower bounds on Private Information Retrieval Protocols have been hard to obtain. The lower bounds that are known either limit the type of query (Goldreich *et al.* 2002; Mann 1998) or are weak (Kerenidis & de Wolf 2004; Wehner & de Wolf 2005).

We assume throughout the paper that the queries sent to each server are the same length and that there are only 2 servers. Consider the case that the answers from the database are linear, i.e., they are an XOR of some subset of the bits of the database. Goldreich *et al.* (2002) show that $\Omega(\frac{n}{2^a})$ bits must be sent to each server where a is the number of bits each server could send back to Alice. The lower bound also holds for randomized protocols with a small probability of error. The multiplicative constant depends on the probability of error. They obtain their results by proving lower bounds on locally decodable codes and then showing how such lower bounds imply lower bounds for PIR's. In the special case of $a = 1$ where Alice simply XORs the bits she gets Chor *et al.* (1998) show that any protocol would require $n-1$ bits sent to each server. They also give a matching upper bound in this model.

In the case that answers are not restricted to be linear, nontrivial lower bounds have only recently been discovered. Kerenidis & de Wolf (2004) show that at least $\Omega(n/2^{5a})$ bits must be sent to each server. This has been improved to $\Omega(n/2^{2a})$ by Wehner & de Wolf (2005). In the case $a = 1$ Kerenidis & de Wolf (2004) show that at least $(1-H(11/14))n-4 \sim 0.25n$ bits are required. Their proof first converts a 2-server randomized protocol to a 1-server quantum protocol and then they show lower bounds on the quantum protocol. Hence their lower bounds hold for randomized protocols that allow a small probability of error. They also used locally decodable codes.

In this paper we obtain a lower bound of $n - 2$ for 2-server deterministic error-free PIR schemes with the assumption that the answers are 1-bit long. This nearly matches the $n - 1$ upper bound of Chor, Kushilevitz, Goldreich & Sudan (1998).

We avoid the quantum techniques used by Kerenidis & de Wolf (2004). Rather our proof builds on classical tools developed by Yao (1990) and Fortnow & Szegedy (1992) for studying locally-random reductions, a complexity-theoretic tool for information hiding that predates private information retrieval.

2. The Lower Bound

In this section we formally define the model and state and prove our main result.

DEFINITION 2.1. *A 2-server 1-round r-random bit PIR for databases of size n with m-bit queries and a-bit answers is a tuple $(q_1, q_2, a_1, a_2, \phi)$ such that the following hold.*

(i) $q_j : [n] \times \{0,1\}^r \to \{0,1\}^m$. *This is the query sent to server j. The distribution of $q_j(i,\rho)$ is independent of i (this ensures privacy).*

(ii) $a_j : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^a$. *This is the response server j gives if the database is $x \in \{0,1\}^n$ and he sees query $\mu \in \{0,1\}^m$.*

(iii) $\phi : [n] \times \{0,1\}^r \times \{0,1\}^m \times \{0,1\}^m \times \{0,1\}^a \times \{0,1\}^a \to \{0,1\}$. *This is how Alice puts together the information she has received. Say she wants to know x_i. If the random string is $\rho \in \{0,1\}^r$ and the queries are q_1, q_2, and she gets back a-bit strings b_1 and b_2 then Alice computes $x_i = \phi(i, \rho, q_1, q_2, b_1, b_2)$. (Note that since q_1 and q_2 are functions of i, ρ we could have defined ϕ to be a function of just (i, ρ, b_1, b_2); however, making q_1, q_2 explicit inputs has notational advantages.)*

NOTE 2.2. *Note that our PIR's always give the correct answer.*

Throughout this paper we will use the mythical character Alice. Alice is computationally unbounded and knows the protocol but she does not know x. We will be concerned with what she can and cannot deduce from other information she is given.

Assume that $(q_1, q_2, a_1, a_2, \phi)$ is a 2-server 1-round r-random bit PIR for databases of size n with m-bit queries and 1-bit answers. Imagine that Alice wants to find x_i, has random string ρ, and has found out $a_1(x, q_1(i,\rho))$. It is possible that $a_2(x, q_2(i,\rho))$ is not needed. This would happen if $a_2(x, q_2(i,\rho)) = 0$ and $a_2(x, q_2(i,\rho)) = 1$ yield the same value for x_i. If this happens then we say that $i, \rho, a_1(x, q_1(i,\rho))$ *set* x_i. It is also possible that $a_2(x, q_2(i,\rho))$ is crucial. In this case, if Alice happened to know x_i she could determine $a_2(x, q_2(i,\rho))$. In this case we say that $i, \rho, a_1(x, q_1(i,\rho))$ and x_i *force* $a_2(x, q_2(i,\rho))$. Either way is a win. The next definition and lemma formalize this notion.

For the next definition and the two lemmas following it let $(q_1, q_2, a_1, a_2, \phi)$ be a 2-server 1-round r-random bit PIR for databases of size n with m-bit queries and 1-bit answers.

DEFINITION 2.3. *Let $i \in [n]$, $\rho \in \{0,1\}^r$, and $x \in \{0,1\}^n$.*

(i) The values of $i, \rho, a_1(x, q_1(i,\rho))$ set x_i if

$$\phi(i, \rho, q_1(i,\rho), q_2(i,\rho), a_1(x, q_1(i,\rho)), 0) =$$
$$\phi(i, \rho, q_1(i,\rho), q_2(i,\rho), a_1(x, q_1(i,\rho)), 1).$$

Note that if Alice knows i, ρ, and $a_1(x, q_1(i,\rho))$ then she knows x_i. This is a win. The statement "the values of $i, \rho, a_2(x, q_2(i,\rho))$ set x_i" can be defined similarly.

(ii) We say the values of $i, \rho, a_1(x, q_1(i,\rho))$, and x_i force $a_2(x, q_2(i,\rho))$ if

$$\phi(i, \rho, q_1(i,\rho), q_2(i,\rho), a_1(x, q_1(i,\rho)), 0) \neq$$
$$\phi(i, \rho, q_1(i,\rho), q_2(i,\rho), a_1(x, q_1(i,\rho)), 1).$$

Note that if Alice knows $i, \rho, a_1(x, q_1(i,\rho))$ and x_i then she knows $a_2(x, q_2(i,\rho))$. This is also a win. The statement "the values of $i, \rho, a_2(x, q_2(i,\rho))$, and x_i force $a_1(x, q_1(i,\rho))$" can be defined similarly.

In order for the definitions of *set* and *force* to work it is crucial that the protocol only allows 1-bit answers. The next Lemma uses these notions. It is the only place we use that the answers are 1 bit long. Any attempt to extend our proof to 2 or more bits will have to get around this obstacle. The definitions also use the fact that the protocol is always correct.

The following lemma follows from the Definition 2.3

LEMMA 2.4. *Let $i \in [n]$, $\rho \in \{0,1\}^r$, and $x \in \{0,1\}^n$. Then both of the following hold:*

(i) Either $i, \rho, a_1(x, q_1(i,\rho))$ set x_i or $i, \rho, a_1(x, q_1(i,\rho))$, and x_i force $a_2(x, q_2(i,\rho))$.

(ii) Either $i, \rho, a_2(x, q_2(i,\rho))$ set x_i or $i, \rho, a_2(x, q_2(i,\rho))$, and x_i force $a_1(x, q_1(i,\rho))$.

NOTATION 2.5.

(i) Let ORD_1 (ORD_2) be a set of ordered pairs of queries to the first (second) server and the answers to those queries. The phrase 'Alice can deduce x_i from ORD_1, ORD_2, and i' means that Alice, who has unlimited power and access to the protocol, can determine a value $b \in \{0,1\}$ such that $x_i = b$ is consistent with her data while $x_i \neq b$ is not.

(ii) *We can define a similar notion of deduce for other information Alice may have. For example, it is possible that if Alice knows some x_i and some query answers she can deduce other query answers (see Definition 2.3).*

We will need to keep track of queries and answers. We will also need to keep track of where the queries came from. So we have the following notation.

NOTATION 2.6.

(i) *When we use the tuple $(j, i, \rho, q) \in \{1,2\} \times [n] \times \{0,1\}^r \times \{0,1\}^m$ it is implicit that $q = q_j(i, \rho)$. We are interpreting this tuple as saying that if Alice wants to determine x_i, and the random string is ρ, then she sends q to Server j.*

(ii) *$Q(j, i, \rho, q) = (j, q)$. The function Q extracts the query.*

(iii) *$ANS(j, i, \rho, q, x) = (j, q, a_j(x, q))$. The function ANS extracts the query and its answer.*

(iv) *If $S \subseteq \{1,2\} \times [n] \times \{0,1\}^r \times \{0,1\}^m$ then*

$$\begin{aligned} Q(S) = &\ \{q \mid (j, i, \rho, q) \in S\} \\ ANS(S, x) = &\ \{(j, q, a_j(x, q)) \mid (\exists \rho)[(j, i, \rho, q) \in S]\} \end{aligned}$$

Note that $Q(S)$ and $ANS(S, x)$ are not multisets.

LEMMA 2.7. *Let $x \in \{0,1\}^n$ and let $i_0 \in [n]$. Let S^1, S^2 be such that, for $j = 1, 2$, $S^j \subseteq \{j\} \times \{i_0\} \times \{0,1\}^r \times \{0,1\}^m$. Assume that Alice knows the set $ANS(S^1 \cup S^2, x)$ but cannot deduce x_{i_0}. We define T^1, T^2 so that, intuitively, they contains queries that Alice cannot know the answer to given that she does not know x_{i_0}. Formally let T^1 and T^2 be the following sets.*

$$\begin{aligned} T^1 &= \{(1, i_0, \rho, q) \mid (\exists q')[(2, i_0, \rho, q') \in S^2]\}; \\ T^2 &= \{(2, i_0, \rho, q) \mid (\exists q')[(1, i_0, \rho, q') \in S^1]\}; \end{aligned}$$

Then

(i) *If Alice knows i_0 and x_{i_0} then she can deduce $ANS(T^1 \cup T^2, x)$.*

(ii) *$|T^1| = |S^2|$ and $|T^2| = |S^1|$.*

(iii) *$|(S^1 \cup T^1) \cup (S^2 \cup T^2)| = 2|S^1 \cup S^2|$.*

PROOF. 1) Let $(1, i_0, \rho, q) \in T^1$. By definition of T^1, $(\exists q')[(2, i_0, \rho, q') \in S^2]$. By Lemma 2.4 either

1. $i_0, \rho', a_1(x, q_1(i_0, \rho'))$ *set* x_{i_0}, or
2. $i_0, \rho', a_1(x, q_1(i_0, \rho))$, and x_{i_0} *force* $a_2(x, q_2(i_0, \rho'))$.

Since Alice cannot deduce x_{i_0} from $ANS(S^1 \cup S^2, x)$ case a cannot happen. Hence case b happens. Therefore if Alice knows x_{i_0} and $ANS(S^1 \cup S^2, x)$ then she can deduce $a_1(x, q_1(i_0, \rho)) = a_1(x, q)$. A similar proof holds for $(2, i_0, \rho, q) \in T^2$.

2) There is a bijection between T^2 and S^1: map $(2, i_0, \rho, q)$ to $(1, i_0, \rho, q') \in S^1$ such that $q' = q_1(i_0, \rho)$.

3) We need that $S^1 \cap T^1 = \emptyset$. Assume, by way of contradiction, that $(1, i_0, \rho, q) \in S^1 \cap T^1$. Since $(1, i_0, \rho, q) \in T^1$, $(2, i_0, \rho, q) \in S^2$. Since $(1, i_0, \rho, q) \in S^1$ there exists $(2, i_0, \rho, q) \in S^2$, Alice can deduce x_{i_0}. This is a contradiction. □

THEOREM 2.8. *Let $r \in \mathsf{N}$. Any 2-server 1-round r-random bit PIR for databases of size n with m-bit queries and 1-bit answers must have $m \geq n - 2$.*

PROOF. Let $(q_1, q_2, a_1, a_2, \phi)$ be a 2-server 1-round r-random bit PIR for databases of length n with m-bit queries and 1-bit answers. We can assume $r \geq m$ by padding.

Let $M^1(i), M^2(i) \subseteq \{1, 2\} \times [n] \times \{0, 1\}^r \times \{0, 1\}^m$ be defined as follows.

$$\begin{aligned} M^1(i) = &\ \{(1, i, \rho, q) \mid \rho \in \{0, 1\}^r\} \\ M^2(i) = &\ \{(2, i, \rho, q) \mid \rho \in \{0, 1\}^r\} \end{aligned}$$

By privacy, for all $i \in [n]$,

$$\begin{aligned} Q(M^1(1)) = &\ Q(M^1(i)) \\ Q(M^2(1)) = &\ Q(M^2(i)) \end{aligned}$$

By privacy, for every $i_0 \in [n]$, there is a 1-1 onto map F_{i_0} with domain

$$\{(1, 1, \rho, q) \mid q_1(1, \rho) = q\} \cup \{(2, 1, \rho, q) \mid q_2(1, \rho) = q\}$$

and range

$$\{(1, i_0, \rho, q) \mid q_1(i_0, \rho) = q\} \cup \{(2, i_0, \rho, q) \mid q_2(i_0, \rho) = q\}$$

The map is defined by

$F_{i_0}(1,1,\rho,q) = (1,i_0,\rho',q)$ where $q_1(i_0,\rho') = q$,
and
$F_{i_0}(2,1,\rho,q) = (2,i_0,\rho',q)$ where $q_2(i_0,\rho') = q$,

Note that if Alice knew $ANS(M^1(1) \cup M^2(1), x)$ then she would know x. In the construction below we will keep track of which queries Alice knows he answer to by keeping track of which subset of $M^1(1) \cup M^2(1)$ Alice knows the answers for.

Fix ρ. For every $i \in [n]$ there exists ρ', ρ'' such that $q_1(1,\rho) = q_1(i,\rho')$ and $q_2(1,\rho) = q_2(i,\rho'')$.

We exhibit an injection $f : \{0,1\}^n \to \{0,1\}^{m+2}$, hence we obtain $n \leq m+2$, so $m \geq n-2$. The proof that f is an injection will follow easily from the fact that from $f(x)$ and the protocol Alice can reconstruct x.

Since $|M^1(1)| = 2^r$ and the number of possible queries is $|\{0,1\}^m| = 2^m$ there must be a query that occurs with multiplicity 2^{r-m}. Let μ_0 be that query. Let $\rho_1, \ldots, \rho_{2^{r-m}}$ be distinct strings such that

$$\mu_0 = q_1(1,\rho_1) = q_1(1,\rho_2) = \cdots = q_1(1,\rho_{2^{r-m}}).$$

We describe a process for generating a (short) string we call $ADVICE$ that will begin with $a_1(x,\mu_0)$ but then have several bits of x. From $ADVICE$ we will be able to reconstruct the entire string x. We will end up taking $f(x)$ to be $ADVICE$ padded with 0's to make it the right length.
Intuition: At the end of stage ℓ we will have the following.

1. A set $I_\ell \subseteq [n]$. If $i \in I_\ell$ then from $ADVICE_\ell$ Alice can deduce x_i.

2. A string $ADVICE_\ell$ which is (aside from the first bit) the concatenation of x_i for some of the $i \in I_\ell$. The ideas is that $i \in I_\ell$ if either x_i is in $ADVICE_\ell$, or we can use the bits in $ADVICE_\ell$ to deduce the answers to queries that are needed to compute x_i.

3. A set $S^1_\ell \subseteq \{1\} \times \{1\} \times \{0,1\}^r \times \{0,1\}^m$, and a set $S^2_\ell \subseteq \{2\} \times \{1\} \times \{0,1\}^r \times \{0,1\}^m$.

4. Given $ADVICE_\ell$, the protocol, and the construction so far, Alice will be able to deduce the following.

   (a) The set $ANS(S^1_\ell \cup S^2_\ell, x)$.

   (b) For every $i \in I_\ell$, x_i.

These answers will enable Alice to deduce some values of x_i. If x_{i_0} cannot be deduced then adding x_{i_0} to the advice will increase the number of queries in $M^1(1) \cup M^2(1)$ for which Alice can deduce the answers. We will end up with $|S^1_{\ell+1} \cup S^2_{\ell+1}| = 2|S^1_\ell \cup S^2_\ell|$.

We now give the formal construction.

1. Let $ADVICE_0 = a_1(x, \mu_0)$. Throughout the construction $ADVICE_\ell \in \{0,1\}^*$ will be $a_1(x, \mu_0)$ followed by a string of bits that represent particular x_i values. We do not need to put i's into the advice as Alice can deduce them from the construction and $ADVICE$.

2. Let S^1_0 be the set $\{(1,1,\rho_1,\mu_0),(1,1,\rho_2,\mu_0),\cdots,(1,1,\rho_{2^{r-m}},\mu_0)\}$. Let $S^2_0 = \emptyset$.

3. Let $I_0 = \emptyset$. Throughout the construction $I_\ell \subseteq [n]$ will be the set of indices i such that Alice can deduce x_i from knowing $ANS(S^1_\ell \cup S^2_\ell, x)$.

4. Assume S^1_ℓ, S^2_ℓ have been constructed and $I_\ell \neq [n]$. Let i_0 be the least element of $[n] - I_\ell$.

   (a)
   $$ADVICE_{\ell+1} = ADVICE_\ell \cdot x_{i_0}.$$

   (b) We will now add elements to S^1_ℓ and S^2_ℓ to obtain $S^1_{\ell+1}$ and $S^2_{\ell+1}$. Let $S^1 = F_{i_0}(S^1_\ell)$ and $S^2 = F_{i_0}(S^2_\ell)$. Note that Alice knows $ANS(S^1 \cup S^2, x)$ but cannot deduce x_{i_0}. Hence S^1, S^2, i_0 satisfy the premise of Lemma 2.7. Let

   $$\begin{aligned} T^1 &= \{(1,i_0,\rho,q) \mid (\exists q',\rho')[(2,i_0,\rho,q') \in S^2]\}; \\ T^2 &= \{(2,i_0,\rho,q) \mid (\exists q',\rho')[(1,i_0,\rho,q') \in S^1]\}; \\ S^1_{\ell+1} &= S^1_\ell \cup F^{-1}_{i_0}(T^1) \\ S^2_{\ell+1} &= S^2_\ell \cup F^{-1}_{i_0}(T^2) \end{aligned}$$

   Note that T_1 and $F_{i_0}(T_1)$ contain the same queries and are of the same size. By Lemma 2.7 Alice can deduce $ANS(T^1 \cup T^2, x)$ from x_{i_0}. Since Alice can already deduce $ANS(S^1_\ell \cup S^2_\ell, x)$, she can now deduce $ANS(S^1_{\ell+1} \cup S^2_{\ell+1}, x)$. Also by Lemma 2.7 $|S^1_{\ell+1} \cup S^2_{\ell+1}| = 2|S^1_\ell \cup S^2_\ell|$.

   (c)
   $$I_{\ell+1} = I_\ell \cup \{x_{i_0}\} \cup \{i \mid (\exists \rho)[q_1(i,\rho) \in Q(S^1_{\ell+1}) \wedge q_2(i,\rho) \in Q(S^2_{\ell+1})]\}$$

5. If $I_\ell = [n]$ then terminate. If $I_\ell \neq [n]$ then set $\ell = \ell + 1$ and goto step 4. Note that if $S^1_\ell \cup S^2_\ell = M^1(1) \cup M^2(1)$ then $I_\ell = [n]$ and the construction will terminate.

Since $|S^1_0 \cup S^2_0| = 2^{r-m}$ and this union doubles with every stage, so $|S^1_\ell \cup S^2_\ell| = 2^{r-m+\ell}$. Let ℓ' be the final value of ℓ. Since $S^1_\ell \cup S^2_\ell \subseteq M^1(1) \cup M^2(1) = 2^{r+1}$, $r - m + \ell' \leq r + 1$ so $\ell' \leq m + 1$. Since $ADVICE$ began with one additional bit, $|ADVICE| \leq \ell' + 1 \leq m + 2$. Let $f(x)$ be $ADVICE$ followed by enough 0's to pad it out to length $m + 2$. This padding does not affect the reconstruction of x from $f(x)$ since the advice produced for different x's is prefix free. □

3. Open Problems

Chor, Kushilevitz, Goldreich and Sudan (Chor *et al.* 1998) showed that, there is a 2-server 1-round n-random bit PIR for databases of size n with $n - 1$ bit queries and 1-bit answers. By combining this with a general communication balancing technique (also from (Chor *et al.* 1998)) one can obtain the following:

THEOREM 3.1. *Fix $n \in \mathsf{N}$. Let a be such that $a < n$. There exists a 2-server 1-round $(\lceil n/a \rceil - 1)$-random bit PIR for databases of size n with $(\lceil n/a \rceil - 1)$-bit queries and a-bit answers.*

Our lower bound showed that this upper bound is tight in the $a = 1$ case up to an additive constant. It is an open question to show this for all constant a or even for $a = 2$.

Acknowledgements

We would like to thank Jonathan Katz for pointing out that our original proof could be rephrased in terms of simple combinatorics rather than Kolmogorov Theory. We would also like to thank Ronald de Wolf for helpful commentary and updates on his paper with Kerenidis. Thanks to Umesh Vazirani and Stephanie Wehner for helpful discussions and Nan Wang and the anonymous referee for proofreading.

References

ANDRAIS AMBAINIS (1997). Upper Bound on the Communication Complexity of Private Information Retrieval. In *Proceedings of the 24th International Colloquium on Automata, Languages and Programming ICALP 1997,* Bologna, Italy, 401–407.

Amos Beimel, Yuval Ishai, Eyal Kushilevitz & Jean-Francois Rayomnd (2002). Breaking the $O(n^{1/(2k-1)})$ barrier for information-theoretic private information retrieval. In *Proceedings of the 43nd Annual IEEE Symposium on Foundations of Computer Science,* Vancouver, Canada, 261–270.

Christian Cachin, Silvio Micali & Markus Stadler (1999). Computationally private information retrieval with polylog communication. In *EUROCRYPT99*, 402–414.

Benny Chor & Niv Gilboa (1997). Computationally Private Information Retrieval. In *Proceedings of the Twenty-ninth Annual ACM Symposium on the Theory of Computing,* El Paso TX, 304–313.

Benny Chor, Eyal Kushilevitz, Oded Goldreich & Madhu Sudan (1998). Private Information Retrieval. *Journal of the ACM* **45**, 965–981. Earlier version in FOCS 95.

Lance Fortnow & Mario Szegedy (1992). On the Power of Two-Local Random Reductions. *Information Processing Letters* **44**, 303–306.

William Gasarch (2004). A survey on private information retrieval. *Bulletin of the European association of theoretical computer science(BEATCS)* **82**, 84–102. Also see website on this topic: http://www.cs.umd.edu/gasarch/pir/pir.html.

Oded Goldreich, Howard Karloff, Leonard Schulman & Luca Trevisan (2002). Lower Bounds for Linear Local Decodable Codes and Private Information Retrieval Systems. In *Proceedings of the 17th IEEE Conference on Complexity Theory,* Montreal, Canada, 175–183. IEEE Computer Society Press. Updated version on Goldreich's website.

Iordanis Kerenidis & Ronald de Wolf (2004). Exponential Lower Bound for 2-Query Locally Decodable Codes. *Journal of Computer and System Sciences* 395–420. Earlier version in STOC03. E-version at http://arxiv.org/abs/quant-ph/0208062.

Eyal Kushilevitz & Rafail Ostrovsky (1997). Replication Is Not Needed: Single Database, Computationally-Private Information Retrieval (Extended Abstract). In *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science,* Miami Beach FL, 364–373.

Eran Mann (1998). *Private access to distributed information.* Ph.D. thesis, Technion – Israel Institute of Technology, Haifa. Masters Thesis.

S. Wehner & R. de Wolf (2005). Improved Lower Bounds for Locally Decodable Codes and Private Information Retrieval. In *Proceedings of the 32nd ICALP*, volume 3580 of *LNCS*, 1424–1436. Arxiv.org/abs/quant-ph/0403140.

Andrew Yao (1990). An Application of Communication Complexity to Cryptography. Lecture given at DIMACS Workshop on Structural Complexity and Cryptography.

Manuscript received 30 October 2003

Richard Beigel
Dept. of Computer and Information Sciences
Temple University
1805 N. Broad St
Philadelphia, PA 19122
professorb@gmail.com
http://www.cis.temple.edu/~beigel/long.html

Lance Fortnow
Department of Computer Science
University of Chicago
1100 E. 58th St., Chicago, IL 60637
fortnow@cs.uchicago.edu
http://people.cs.uchicago.edu/~fortnow/

William Gasarch
University of Maryland
Dept. of Computer Science and Institute for Advanced Computer Studies
College Park, MD 20742
gasarch@cs.umd.edu
http://www.cs.umd.edu/~gasarch

ECCC ISSN 1433-8092
http://eccc.hpi-web.de/