One-way Functions are Essential for Single-Server Private Information Retrieval

Amos Beimel [*] Yuval Ishai [†] Eyal Kushilevitz [‡] Tal Malkin [§]

Abstract

Private Information Retrieval (PIR) protocols allow a user to read information from a database without revealing to the server storing the database which information he has read. Kushilevitz and Ostrovsky [23] construct, based on the quadratic residuosity assumption, a single-server PIR protocol with small communication complexity. Cachin, Micali, and Stadler [5] present a single-server PIR protocol with a smaller communication complexity, based on the (new) Φ-hiding assumption.

A major question, addressed in the present work, is what assumption is the minimal assumption necessary for the construction of single-server private information retrieval protocols with small communication complexity. We prove that if there is a (0-error) PIR protocol in which the server sends less than n bits then one-way functions exist (where n is the number of bits in the database). That is, even saving one bit compared to the naive protocol, in which the entire database is sent, already requires one-way functions. The same result holds (but requires more work) even if we allow the retrieval to fail with probability of at most $1/(8n)$. Moreover, similar results hold even if we allow constant probability of error. For example, we prove that if there is a PIR protocol with error $1/4$ and communication complexity less than $n/10$ bits, then one-way functions exist.

*Division of Engineering and Applied Sciences, Harvard University, 40 Oxford st., Cambridge, MA 02138. E-mail: beimel@deas.harvard.edu. Supported by grants ONR-N00014-96-1-0550 and ARO-DAAL03-92-G0115.

†Computer Science Department, Technion, Haifa 32000, Israel. E-mail: yuvali@cs.technion.ac.il. Part of this work was done while visiting IBM T.J. Watson Research Center.

‡IBM T.J. Watson Research Center, and Computer Science Department, Technion, Haifa, Israel. E-mail: eyalk@watson.ibm.com and eyalk@cs.technion.ac.il. Supported in part by the Mitchell-Schoref program at the Technion.

§Laboratory for Computer Science, Massachusetts Institute of Technology, 545 Technology sq., Cambridge, MA 02139. E-mail: tal@theory.lcs.mit.edu. Supported by DARPA grant DABT63-96-C-0018. Part of this work was done while visiting IBM T.J. Watson Research Center.

1 Introduction

Protecting the privacy of a user searching in a database is an important task. Specifically, hiding the information that the user is interested in is of much interest. For example, an investor might want to know the value of a certain stock in the stock-market without revealing the identity of this stock. Private information retrieval (PIR) protocols, introduced by Chor, Goldreich, Kushilevitz, and Sudan [8], allow a user to access a database without the server storing the database having any information on which records the user read. For sake of concreteness, the database is modeled as an n bit string x. The user has some index i and he is interested in privately retrieving the value of x_i. The simplest PIR protocol is for the user to read the entire database (although he is typically interested only in a small portion of the database). This solution is very expensive in terms of communication complexity. However, if the server is not allowed to gain any information about the index i (i.e., information theoretic privacy) then this solution is optimal [8].

To overcome this problem, Chor et al. [8] suggested that the user accesses replicated copies of the database kept on different servers. They suggested protocols in which each server gets no information on the bit the user reads. These protocols include (among others) a 2-server protocol with communication complexity of $O\left(n^{1/3}\right)$ bits, and a k-server protocol with communication complexity of $O\left(n^{1/k}\right)$ bits for a constant k. This was improved by Ambainis [2] to $O\left(n^{1/(2k-1)}\right)$ bits (see also [21]). However, in all these protocols it is assumed that the servers do not communicate with each other.

A different approach to reduce the communication is to limit the power of the servers; that is, to assume that each server is a probabilistic polynomial-time machine. Chor and Gilboa [6] prove that if one-way functions exist then for every constant $\epsilon > 0$ there exists a 2-server protocol in which the communication is $O(n^{\epsilon})$ bits. Kushilevitz and Ostrovsky [23] prove that one server suffices; under the quadratic residuosity assumption they construct, for every constant $\epsilon > 0$, a *single* server protocol with communication complexity of $O(n^{\epsilon})$ bits. Moreover, under a stronger assumption regarding the hardness of the quadratic residuosity problem, the communication complexity of this protocol becomes $2^{O(\sqrt{\log n \log\log n})}$ bits. Mann [25] generalizes the construction of [23] to use any trapdoor predicate with certain homomorphic properties. Specifically, such predicates exist under the decisional Diffie-Hellman assumption and under the assumption that it is hard to approximate the shortest vector in a lattice. Cachin, Micali, and Stadler [5] present a protocol with poly-logarithmic communication complexity, based on a new intractability assumption called the Φ-hiding assumption.

A major question addressed in the present work is what assumption is the minimal assumption necessary for the construction of single-server private information retrieval protocols with small communication complexity. We prove that if there is a (0-error) protocol in which the server sends less than n bits then one-way functions exist (where n is the number of bits in the database). That is, even saving one bit compared to the naive protocol, in which the entire database is sent, already requires one-way functions. Similar results hold (but require more work) even if we allow the retrieval to fail with some small probability ϵ. More specifically, we prove that if there is a PIR protocol with error $\epsilon \leq 1/(8n)$ and communication complexity less than n bits then one-way functions exist. The same result holds if there is a protocol with constant error $\epsilon < 1/2$ and communication complexity of at most $(1 - H(\epsilon)) \cdot n - 1$ (where $H(\epsilon)$ is the binary entropy function). For example, if there is a protocol with error $\epsilon \leq 1/4$ and communication complexity of at most $n/10$ bits then one-way functions exist.

We present two different proofs of our main result that private information retrieval protocols with small communication imply the existence of one-way functions. The first proof shows that PIR protocols with small communication complexity can be used to construct bit-commitment protocols. By Impagliazzo and Luby [19], the existence of bit-commitment protocols implies the existence of one-way functions. The second proof is a direct proof showing explicitly how to construct one-way functions from PIR protocols. The second proof is somewhat stronger as it works also for PIR protocols with larger communication complexity (and with reconstruction errors). Together with the previously known fact that one-way functions imply the existence of bit-commitment protocols [26, 18], our second proof (indirectly) shows how to construct bit-commitment protocols from PIR protocols. However, the direct construction given in the first proof is much more efficient.

Our result continues a series of works showing that the existence of one-way functions is a minimal assumption for many cryptographic primitives. For example, each of the following protocols imply the existence of one-way functions: private-key encryption, identification, and bit commitment (Impagliazzo and Luby [19]), Digital Signatures (Rompel [30]), and non-trivial zero-knowledge protocols (Ostrovsky and Wigderson [28]). Furthermore, the existence of two distribution ensembles that are statistically far but are indistinguishable implies the existence of one-way functions (Goldreich [15]).

Subsequent Work: Following our work, Di Crescenzo, Malkin, and Ostrovsky [12] prove a stronger result, showing that in fact the existence of single-server PIR protocols with communication complexity of less than n bits implies the existence of oblivious transfer. Combined with a result of Impagliazzo and Rudich [20], this gives evidence that single-server PIR protocols with small communication complexity are *not* likely to be implemented based on one-way functions alone. We emphasize that it is not clear whether oblivious transfer is *sufficient* to construct PIR with short communication. It remains as an intriguing open problem whether it is possible to construct private information retrieval protocols with short communication based on the existence of one-way permutations with trapdoor.

Related Work: First, we note that PIR protocols are related to instance hiding schemes [1, 3, 4] (see [8] for a discussion). There are also various extensions of PIR in the literature. In particular, protocols for private retrieval and storage were presented in [27]. PIR Protocols which also protect the database, by restricting the user to read only one bit in each invocation, were presented in [14]. Other extensions of PIR protocols are presented in [11, 13].

Organization: In Section 2 we define private information retrieval protocols and one-way functions. In Section 3 we show how to construct bit-commitment protocols from private information retrieval protocols. In Section 4 we give a direct proof that private information retrieval protocols with small communication and without reconstruction errors can be used to construct one-way functions, and in Section 5 we extend this result to private information retrieval protocols with errors in the reconstruction.

2 Preliminaries

2.1 Private Information Retrieval

We define single-server, one-round computationally-private information retrieval protocols. The privacy we require in the definition is computational and requires the following notation and definition. A distribution ensemble $\{X^n\}_{n=1}^{\infty}$ is a sequence of random variables $X^1, X^2, \ldots$.

Definition 2.1 (Indistinguishable Distributions) *Two distribution ensembles $\{X^n\}_{n=1}^{\infty}$ and $\{Y^n\}_{i=n}^{\infty}$ are indistinguishable in polynomial-time (or* indistinguishable *for short) if for every probabilistic polynomial-time algorithm M, every integer $c \geq 1$ and for every sufficiently large n*

$$\left| \Pr_{x \in X^n} [M(x, 1^n) = 1] - \Pr_{y \in Y^n} [M(y, 1^n) = 1] \right| \leq \frac{1}{n^c}.$$

We are ready to define private information retrieval protocols.

Definition 2.2 (PIR) *A Private Information Retrieval (abbreviated PIR) protocol involves two parties: a server $\mathcal{S}$ which holds an n-bit string x (the database) and a user $\mathcal{U}$ who wants to retrieve a bit x_i of the database (where $i \in \{1, ..., n\}$).*

A PIR protocol $\mathcal{P} = (\mathcal{Q}, \mathcal{A}, \mathcal{C})$ consists of three efficient algorithms: a query algorithm $\mathcal{Q}(\cdot,\cdot,\cdot)$, an answering algorithm $\mathcal{A}(\cdot,\cdot)$ and a reconstruction algorithm $\mathcal{C}(\cdot,\cdot,\cdot,\cdot)$. At the beginning of the protocol, the user picks a random string r and computes a query $q = \mathcal{Q}(1^n, i, r)$, where n is the length of the database.[1] *The user then sends the query q to the server. The server responds with an answer $a = \mathcal{A}(q, x)$ (the answer is a function of the query and the database; without loss of generality the server is deterministic). Finally, the user computes the bit x_i by applying the reconstruction algorithm $\mathcal{C}(1^n, i, r, a)$. The input of each algorithm contains the length of the database n (in an unary encoding) and its running time is polynomial in n. (This implies that the lengths of r, q and a are polynomial in n.) There are two requirements from the protocol:*

Perfect Correctness Requirement *The user always computes the correct value of x_i. Formally, for every index $i \in \{1, ..., n\}$, every random string r, and every database $x \in \{0,1\}^n$*

$$\mathcal{C}(1^n, i, r, \mathcal{A}(\mathcal{Q}(1^n, i, r), x)) = x_i.$$

Privacy Requirement *The server has no information about the bit that the user tries to retrieve: For every two sequences of indices $\{i_n\}_{n=1}^{\infty}$ and $\{j_n\}_{n=1}^{\infty}$, where $1 \leq i_n, j_n \leq n$, the distribution ensembles $\{\mathcal{Q}(1^n, i_n, \cdot)\}$ and $\{\mathcal{Q}(1^n, j_n, \cdot)\}$ are indistinguishable in polynomial-time.*

By the *communication complexity* of a PIR protocol we refer in this work to the size of the server's answer alone. In our context, since we prove lower bounds, this only strengthens the results.

[1] In current PIR protocols [23, 5] the user's queries are computed in *expected* polynomial-time. However, the query computation can be easily made polynomial in the worst case without compromising the correctness, by making the user send some fixed valid query pointing to i in the case that the query generation fails (the probability of which can be made exponentially small).

Definition 2.3 (Communication Complexity) *We say that the communication complexity of a PIR protocol $\mathcal{P} = (\mathcal{Q}, \mathcal{A}, \mathcal{C})$ is (bounded by) $s(n)$ bits if for every n, every index $i \in \{1, ..., n\}$, every random string r, and every database $x \in \{0,1\}^n$ the length of the message $\mathcal{A}(\mathcal{Q}(1^n, i, r), x)$ sent by the server is at most $s(n)$.*

Finally, although by default we consider single-round PIR protocols, Definitions 2.2 and 2.3 generalize in a natural way to multi-round PIR protocols. All the results presented in this work carry on to the multi-round case.

2.2 One-way functions

Loosely speaking, a one-way function is a function that can be easily computed but is hard to invert on an image of a random input. The notion of one-way functions used in cryptographic applications, also called *strong* one-way functions, requires that no algorithm can invert the function on any inverse polynomial fraction of the input space. In this paper we will work with *weak* one-way functions; that is, functions that are hard to invert for at least an inverse polynomial fraction of the input space (but might still be easy to invert on most inputs). Yao [31] proved that if weak one-way functions exist then strong one-way functions exist. Thus, to prove the existence of strong one-way functions it suffices to prove the existence of weak one-way functions. Therefore, we only formalize the notion of weak one-way functions. For a definition of strong one-way functions and their equivalence to weak one-way functions see, e.g., [16, 24].

Definition 2.4 (Weak One-way Functions) *A function $f : \{0,1\}^* \to \{0,1\}^*$ is a weak one-way function if the following two conditions hold:*

Easy to compute *There exists a deterministic polynomial-time algorithm A that for every input x outputs $f(x)$ (i.e., $A(x) = f(x)$).*

Slightly Hard to invert *There is a constant c such that for every probabilistic polynomial-time algorithm M, and for every sufficiently large n*

$$\Pr\left[M(f(x), 1^n) \notin f^{-1}(f(x))\right] \geq \frac{1}{n^c},$$

where the probability is taken over x chosen uniformly from $\{0,1\}^n$ and over the random choices of M.

3 Bit Commitment from PIR

In this section we describe a simple construction of a bit-commitment protocol based on any (single-server) PIR protocol $\mathcal{P}$ with small communication. Since commitment is known to imply one-way functions [19], such a construction is sufficient for obtaining the main qualitative result of this work. However, *direct* proofs of *tighter* results will be given in subsequent sections; we shall thus omit full proofs from this section, and will not attempt to derive the strongest bounds possible.

Bit Commitment. Informally, a *commitment* protocol may be viewed as a cryptographic implementation of a sealed opaque envelope: selected information can be kept secret by storing it inside such an envelope, and when opening the envelope later it is only possible to reveal the stored information (and no other information). More formally, bit commitment is a protocol between two probabilistic polynomial-time players: a *committer* Alice, holding an input bit b, and a *receiver* Bob. Both players hold a security parameter n as an input. The protocol consists of two phases: an interactive *commit* phase, at the end of which Bob holds some (encrypted) representation of b, and a *decommit* phase, in which Alice sends Bob a single decommitment string dec, and Bob either outputs a bit b' or rejects. A commitment protocol (Alice , Bob) should satisfy the following three properties:

Correctness When both players are honest, Bob outputs the correct bit b with overwhelming probability;

Security If Alice is honest, then any probabilistic polynomial-time (possibly dishonest) Bob* cannot learn the value of the bit b during the commit phase (i.e., his view keeps b semantically secure [17]); and

Binding For any probabilistic polynomial-time (possibly dishonest) Alice*, only with negligible probability can Alice* "cheat" by coming up, following the commit phase, with decommitment strings dec_0, dec_1 that are opened by Bob as different bits.

We refer the reader to, e.g., [19] for a more formal definition.

We first describe a construction of a *weak bit-commitment protocol*, in which the above binding property is replaced by the following, weaker, property:

Weak binding For any probabilistic polynomial-time Alice*, the probability that Alice* successfully cheats (in the sense defined above) is at most $1 - 1/p(n)$ for some polynomial $p(n) > 0$.

The weak bit-commitment protocol, based on the given PIR protocol $\mathcal{P}$, is described in Figure 1. The proof of the security of the protocol utilizes the high (two-party) communication complexity of the inner product function IP, defined for $x, y \in \{0,1\}^n$ by:

$$\mathrm{IP}(x,y) \stackrel{\mathrm{def}}{=} \sum_{i=1}^{n} x_i \cdot y_i \bmod 2.$$

(See, e.g., [22] for background on communication complexity.)

Claim 3.1 *Let $\mathcal{P}$ be a (multi-round) PIR protocol in which the server communicates at most $n/2$ bits. Then,* Weak-Commit$_{\mathcal{P}}$ *is a weak bit-commitment protocol.*

Protocol Weak-Commit$_{\mathcal{P}}$

Commit phase

1. Alice selects two independent uniformly random strings $x, y \in \{0,1\}^n$, and Bob selects a uniformly random index $i \in \{1, 2, \ldots, n\}$.
2. Alice and Bob execute the PIR protocol $\mathcal{P}$, where Alice simulates the server on the database x, and Bob simulates the user on retrieval index i.
3. Alice sends y and $b \oplus \mathrm{IP}(x,y)$ to Bob (where IP denotes inner product over GF(2)).

Decommit phase

1. Alice sends x as decommitment string.
2. Bob verifies that the string sent by Alice is consistent with the bit he retrieved during the commit phase (otherwise Bob rejects), and recovers b from the strings x, y, and the bit $b \oplus \mathrm{IP}(x,y)$.

Figure 1: A weak bit-commitment protocol based on a low communication PIR protocol $\mathcal{P}$.

Proof sketch: First, it is easy to verify that the protocol satisfies the correctness property.

The weak binding property follows from the privacy of the PIR protocol $\mathcal{P}$. If Alice* successfully cheats by coming up with $x = dec_0$ and $x' = dec_1$ as in the definition, then there must be a position j at which $x_j \neq x'_j$. Any such index j must be *different* from the index i picked by Bob since otherwise Bob rejects the one whose bit is different than the bit that he privately retrieved in the commit phase (which by the privacy of $\mathcal{P}$ is hidden from Alice) . It follows that the cheating probability of Alice* can only be negligibly greater than $1-1/n$, or otherwise the privacy of the underlying PIR protocol $\mathcal{P}$ is compromised.

Finally, to prove the security property we rely on the high communication complexity of the inner product function IP. Suppose that following the commit phase on a random bit b, Bob* can predict the value of b with probability $1/2 + \epsilon$. Then one can obtain from Alice and Bob* a two-party (one-round) deterministic protocol with communication complexity $n/2$, predicting the inner product of two random n-bit input strings, x (held by the first party) and y (held by the second), with probability $1/2 + \epsilon$ (where the probability is taken over x and y chosen uniformly and independently from $\{0,1\}^n$).[2] It follows from lower bounds on the randomized communication complexity of inner product, proved by Chor and Goldreich [7], that in any such protocol with less than $n/2$ bits of communication, the advantage ϵ must be exponentially small (that is, at most $2^{-\Omega(n)}$). $\Box$

Finally, the weak binding property can be strengthened by requiring Alice to independently commit to the same bit b polynomially (say n^2) many times, using WEAK-COMMIT$_{\mathcal{P}}$, and letting Bob output a bit b' only if the same bit b' was successfully decommitted every time (or otherwise reject). It can be shown that such sequential repetition would yield strong binding without compromising the security property. That is, Alice* can cheat successfully in such repetition only if she cheats successfully in each of the polynomially many repetitions, and this happens with exponentially small probability.

Remark 3.2 Our protocol may be viewed as a robust version of a protocol due to Crépeau [10], implementing bit commitment based on Rabin's oblivious transfer primitive [29]. In the commit phase of Crépeau's protocol, the committed bit b is first split by Alice into otherwise-random bits $b_1, \ldots, b_n$ whose exclusive-or is equal to b, and then each bit b_i is obliviously transferred to Bob with some constant reception probability p. However, the security property of this protocol might totally break if Bob is allowed to invoke a PIR protocol $\mathcal{P}$ on the database $b_1 \cdots b_n$. Indeed, even a single answer bit of $\mathcal{P}$ may disclose to Bob the exclusive-or of all data bits, which is equal to b.

4 Perfectly-Correct PIR Implies One-Way Functions

In this section we give a direct proof that if there is a single-server PIR protocol with "short" communication complexity then one-way functions exist. We first outline the idea of the proof. Assume the answer of the server for every query is "short" (less than the length of the database). This means that the user cannot reconstruct the entire database from the server's answer. That is, for every query there is an index that could not generate this query. By the privacy requirement, the query must hide this "inconsistent" index. This intuition suggests that $Q(1^n, i, r)$ is a one-way function. The next example shows that this is not necessarily the case; for some PIR protocols the query function $Q(1^n, i, r)$ can be easily inverted.

Example 4.1 Consider a PIR protocol $\mathcal{P} = (\mathcal{Q}, \mathcal{A}, \mathcal{C})$ in which the distribution $\mathcal{Q}(1^n, i, \cdot)$ is indistinguishable from the uniform distribution. Further assume, without loss of generality, that the length of queries equals the length of the random string. We slightly modify $\mathcal{P}$ into a new protocol $\mathcal{P}' = (\mathcal{Q}', \mathcal{A}', \mathcal{C}')$, in which $\mathcal{Q}'(1^n, 1, r) = r$, $\mathcal{Q}'(1^n, i, r) = \mathcal{Q}(1^n, i, r)$ for $i \neq 1$, and $\mathcal{A}'(q, x) = \langle x_1, \mathcal{A}(q, x)\rangle$. Clearly, the user can always reconstruct the bit x_i and the server does not get any information on i. However, to invert $\mathcal{Q}'(1^n, i, r) = q$ the inverting algorithm only needs to output the pre-image $\langle 1^n, 1, q\rangle$.

The problem in the above example is that we find some pre-image of $Q(1^n, i, r)$, and not necessarily with the index i. We prove that a modification of $\mathcal{Q}$, in which we add the index i to the output of the function, yields a weak one-way function. We assume, without loss of generality, that for input size n the query algorithm $\mathcal{Q}$ uses exactly $m(n)$ random bits for some function $m(n) \geq n$. Nevertheless, $m(n) \leq \text{poly}(n)$ since the running time of $\mathcal{Q}$ is polynomial in n. We define the following function

$$f(i, r) \stackrel{\text{def}}{=} \langle i, \mathcal{Q}(1^n, i, r)\rangle. \tag{1}$$

In the rest of the paper we (mis-)use n as the input length. We claim that f is a weak one-way function. That is, given an index i and a query q generated from it ($q = \mathcal{Q}(1^n, i, r)$), it is hard to find a random string r' such that $q = \mathcal{Q}(1^n, i, r')$, for some inverse polynomial fraction of the pairs i, r.

Formally, let $\mathcal{P} = (\mathcal{Q}, \mathcal{A}, \mathcal{C})$ be a PIR protocol with low communication complexity (less than n bits sent from $\mathcal{S}$ to $\mathcal{U}$). Assume (for contradiction) that f is not a weak one-way function. This implies that there is an algorithm INV which, for infinitely many n's,[3] inverts f with probability at least $1 - 1/(2n^2)$ (where the probability is over the uniform choices of r and i, and the random choices of the inverting algorithm INV). In Figure 2 we describe an algorithm DIST that is used to distinguish between the distributions $\mathcal{Q}(1^n, i_n, \cdot)$ and $\mathcal{Q}(1^n, j_n, \cdot)$ for some index sequences i_n and j_n.

We start by proving, in the next two claims, that Algorithm DIST is significantly more likely to output "ACCEPT" when its input includes, in addition to the query

[2] The two party protocol works as follows: Player I on input x and Player II on input y start by executing Step 2 of the Commit Phase; that is, they run the PIR protocol between Alice and Bob* (where Player I simulates Alice with its input x and Player II simulates Bob* with a randomly chosen i). Then, Player II completes the view of Bob* by adding to it messages as Bob* expects to get in Step 3 of the Commit Phase; that is, the string y (i.e., the input of Player II) and a random bit c. Observe that this gives Bob* exactly the distribution of transcripts that it would get by executing Weak-Commit$_{\mathcal{P}}$ where the committed bit b is random. Finally, since Bob* can predict b with probability $1/2 + \epsilon$ then, by taking the exclusive-or of this prediction with c, Player II can predict $\text{IP}(x, y)$ with the same probability. We get a two-party protocol with communication complexity as that of the *total* communication in the underlying PIR protocol, yet not a deterministic one. However, by a standard averaging argument we can fix a random string for Player II (this includes the choice of i and c as well as the internal random choices of Bob*) so as to get a deterministic protocol that computes $\text{IP}(x, y)$ with probability $1/2 + \epsilon$ (over the choice of x and y). Moreover, once Bob* becomes deterministic then the PIR protocol becomes a one-round protocol, with communication complexity as that sent by the server alone (since Alice can immediately simulate the replies of Bob*), as desired.

[3] In the following we implicitly restrict our attention to n's belonging to such infinite sequence.

Description of DIST

Input: a query q and an index j

($\star$ DIST checks if j is likely to be an index that generated q $\star$)

1. Invert $\langle j, q\rangle$ using INV:
   Let $\langle j', r'\rangle \leftarrow \mathrm{INV}(j, q)$.
2. If $j' = j$ and $\mathcal{Q}(1^n, j', r') = q$
   ($\star$ INV inverted correctly $\star$)

   then output "ACCEPT",

   otherwise output "REJECT".

Figure 2: The distinguishing algorithm DIST.

Protocol TRANSMIT

$\mathcal{S}$'s input: a string $y \in \{0,1\}^n$.

- $\mathcal{U}$ chooses a random string r, computes $q \leftarrow \mathcal{Q}(1^n, i, r)$, and sends q to $\mathcal{S}$ (where i is the fixed index from Claim 4.3).
- $\mathcal{S}$ computes $a \leftarrow \mathcal{A}(q, y)$ and sends a to the user.
- For $j \leftarrow 1$ to n the user $\mathcal{U}$ computes:

  $\langle j', r'\rangle \leftarrow \mathrm{INV}(j, q)$, and

  $y'_j \leftarrow \mathcal{C}(1^n, j', r', a)$

$\mathcal{U}$'s output: $y' \leftarrow \langle y'_1, y'_2, \ldots, y'_n\rangle$

Figure 3: A protocol for $\mathcal{S}$ to transmit y to $\mathcal{U}$.

q, the actual index that generated the query than when its input includes some other index.

Claim 4.2 *Fix an arbitrary index j and choose r uniformly at random. The probability that* DIST *outputs "REJECT" on input $\langle \mathcal{Q}(1^n, j, r), j\rangle$ is at most $1/(2n)$, where the probability is over the uniform choice of r and the random choices of the inverting algorithm* INV.

Proof: Since INV fails to invert $f(j,r)$ with probability at most $1/(2n^2)$ (where the probability is taken over the indices j, the random strings r, and the random choices of INV), then for every j it fails to invert $f(j,r)$ with probability at most $1/(2n)$ (where the probability is taken over the random strings r and the random choices of INV). By its definition, DIST outputs "REJECT" only when INV fails, and the claim follows. □

Claim 4.3 *For any fixed index i there exists an index j such that if r is chosen uniformly at random then the probability that Algorithm* DIST *outputs "REJECT" on input $\langle \mathcal{Q}(1^n, i, r), j\rangle$ is at least $1/n$.*

Proof: Fix an index i and assume towards contradiction that for every j the probability that DIST outputs "REJECT" is less than $1/n$. Under this assumption we construct a deterministic protocol in which for every string $y \in \{0,1\}^n$ the server $\mathcal{S}$ sends one message of length less than n bits and the user $\mathcal{U}$ can reconstruct the entire database y, which is impossible by standard information theoretic considerations. First, we present in Figure 3 a randomized protocol, called TRANSMIT, between the user and the server. Then, this protocol will be modified into a deterministic protocol.

We next analyze the probability that $y' = y$ in Protocol TRANSMIT. By our assumption, for every j the probability that

$$j' = j \text{ and } \mathcal{Q}(1^n, j', r') = q \qquad (2)$$

is greater than $1 - 1/n$, where the probability is over the uniform distribution of r and the random choices of Algorithm INV. Thus, with positive probability (2) holds for *every* $j \in \{1, \ldots, n\}$. But any index j for which (2) holds satisfies

$$y'_j = \mathcal{C}(1^n, j, r', \mathcal{A}(\mathcal{Q}(1^n, j, r'), y))$$

and, by the correctness requirement, $y'_j = y_j$ for every $y \in \{0,1\}^n$. Therefore, with positive probability $y' = y$, for every y. This implies that there exists a choice for r and for the random string used by Algorithm INV such that the protocol succeeds for all the strings y in $\{0,1\}^n$. Fix these random strings to obtain a protocol in which the user is deterministic and has no input, thus his first message can be eliminated. Since we assume that $|\mathcal{A}(q, y)| < n$, there are two strings for which the server sends the same message while the deterministic user $\mathcal{U}$ reconstructs the correct string, contradiction. □

We now prove our main result:

Theorem 4.4 *If there is a single-server PIR protocol in which the server communicates less than n bits then one-way functions exist.*

Proof: Assuming the existence of the PIR protocol $\mathcal{P} = (\mathcal{Q}, \mathcal{A}, \mathcal{C})$ as above, we define in (1) the function f. Then, assuming that f is not a weak one-way function, we construct an algorithm that distinguishes between the distributions $\mathcal{Q}(1^n, i_n, \cdot)$ and $\mathcal{Q}(1^n, j_n, \cdot)$ for some indices i_n and j_n: Fix any i_n and let j_n be the index guaranteed by Claim 4.3. By Claim 4.2 if we choose r uniformly, compute $q = \mathcal{Q}(1^n, j_n, r)$, and run Algorithm DIST with input $\langle q, j_n\rangle$ then DIST outputs "REJECT" with probability at most $1/(2n)$. On the other hand, by Claim 4.3 if we choose r uniformly, compute $q = \mathcal{Q}(1^n, i_n, r)$, and run Algorithm DIST with input

$\langle q, j_n \rangle$ then DIST outputs "REJECT" with probability at least $1/n$. This contradicts the privacy requirement, and therefore the assumption that f is not a weak one-way function is false. $\square$

Remark 4.5 The result can be generalized to multi-round single-server PIR protocols. In this case we define the one-way function as follows:

$$f(x, i, r) \stackrel{\text{def}}{=} \langle i, \mathcal{T}(1^n, x, i, r) \rangle,$$

where x is the n-bit database and $\mathcal{T}(1^n, x, i, r)$ is the entire transcript of the communication exchanged between the user and the server. The communication assumption is that the number of bits sent by the server during the protocol is less than n. [4]

Remark 4.6 Our result implies that the length of the query in any PIR protocol with answer length less than n has to be $\omega(\log n)$. That is, $|\mathcal{Q}(1^n, i, r)| > c \log n$ for every constant c (otherwise, f can be inverted in time $n^{O(c)}$). Notice that this is not the case if there is no restriction on the answer length.

5 Dealing with Reconstruction Errors

In this section we extend our main result to PIR protocols with a small probability that the user reconstructs the value of the bit x_i incorrectly. We prove that if there is a PIR protocol with error probability of at most $1/(8n)$ and communication complexity of less than n bits then one-way functions still exist. The same result holds if there is a PIR protocol with error probability $1/4$ and communication complexity of $n/10$ bits. Notice that if we allow a constant probability of error then we can save a constant multiplicative factor in the communication complexity even in the information theoretic model[5].

We start with a formal definition of PIR protocols with reconstruction errors. In this case, we cannot assume that the server is deterministic by fixing its random string. However, in our results we only consider the number of bits sent by the server, and so the server can get the random coins it needs from the user as part of the query (this clearly does not violate the user's privacy), allowing us to still assume the server is deterministic.

[4] In the multi-round case the deterministic version of the protocol TRANSMIT only succeeds for a 3/4 fraction of the strings $y \in \{0,1\}^n$. The details of the proof of this case are similar to the proof of Theorem 5.5 below.

[5] As an example we describe a protocol with error 1/4 and communication complexity of $n/2$ bits: the user chooses at random $j \in \{0,1\}$ and sends j to the server, which responds by sending the $n/2$ bits $x_{jn/2+1}, x_{jn/2+2}, \ldots, x_{jn/2+n/2}$ to the user. With probability 1/2 the bit x_i is one of these bits; in this case the user always outputs the correct value, otherwise he guesses the value of x_i with probability 1/2, for a total success probability of 3/4.

Definition 5.1 (PIR with Errors) *A PIR protocol $\mathcal{P} = (\mathcal{Q}, \mathcal{A}, \mathcal{C})$ has $\epsilon(n)$-error if for every n, every index $i \in \{1, \ldots, n\}$, and every database x the probability that the user reconstructs an incorrect value for x_i is at most $\epsilon(n)$:*

$$\Pr\left[\mathcal{C}(1^n, i, r, \mathcal{A}(\mathcal{Q}(1^n, i, r), x)) \neq x_i\right] \leq \epsilon(n),$$

where the probability is over r, the random choice of the query algorithm $\mathcal{Q}$.

We next give an example showing that there is a PIR protocol with a very small error in which the function f, defined in Section 4, is *not* a one-way function.

Example 5.2 Let $\mathcal{P} = (\mathcal{Q}, \mathcal{A}, \mathcal{C})$ be any PIR protocol. We define a modified PIR protocol $\mathcal{P}' = (\mathcal{Q}', \mathcal{A}', \mathcal{C}')$ which has the same communication complexity as the original protocol and an exponentially small error. However, the function f defined based on the PIR protocol $\mathcal{P}'$ is not a weak one-way function. The query algorithm $\mathcal{Q}'$ is defined as:

$$\mathcal{Q}'(1^n, i, r'r) = \begin{cases} \mathcal{Q}(1^n, i, r) & \text{if } r' \neq 0^n, \ |r'| = n \\ r & \text{if } r' = 0^n. \end{cases}$$

When $\mathcal{A}'$ gets a query that is syntactically correct, its answer is $\mathcal{A}(\mathcal{Q}'(1^n, i, r'r), x)$, and otherwise $\mathcal{A}'$ and $\mathcal{C}'$ can react arbitrarily. Algorithm $\mathcal{Q}'$ is nearly the same as $\mathcal{Q}$ except for some "atypical" random strings that make it easy to invert f: for every $\langle i, q \rangle$ it holds that $f(i, 0^n q) = \langle i, q \rangle$.

Notice that the inverting algorithm in the above example always outputs an "atypical" random string. If we required the inverting algorithm to output a *random* pre-image of f, then an inverting algorithm that outputs "atypical" random strings would fail. This notion of one-way functions in which no algorithm can find a random pre-image, called *distributional one-way functions*, was defined by Impagliazzo and Luby [19], where they showed that if distributional one-way functions exist then one-way functions exist. We show that f is a distributional one-way function.

Before defining distributional one-way functions, we recall that the statistical distance between two distributions D_1 and D_2 over a domain S is defined as

$$\max_{A \subseteq S} \left| \Pr_{D_1}[x \in A] - \Pr_{D_2}[x \in A] \right|.$$

Definition 5.3 (Distributional One-wayness) *A function f is a distributional one-way function if: (1) given x, it is easy to compute the value $f(x)$, and (2) for some constant $c > 0$ and for every probabilistic polynomial-time algorithm M there is an integer n_0 such that for every $n > n_0$ the statistical distance between the distribution $\langle x, f(x) \rangle$ and the distribution $\langle M(f(x)), f(x) \rangle$ is at least $1/n^c$ where x is chosen uniformly in $\{0,1\}^n$.*

Theorem 5.4 (Impagliazzo and Luby [19]) *If distributional one-way functions exist then one-way functions exist.*

Description of DISTRIBUTIONAL-DIST

Input: a query q and an index j

1. Invert j, q: let $\langle j', r'\rangle \leftarrow$ D-INV(j, q).
2. Check if $j' = j$ and $\mathcal{Q}(1^n, j', r') = q$ ($\star$ D-INV inverted correctly $\star$)
3. Choose a random y as the database, and check if
$$\mathcal{C}(1^n, j', r', \mathcal{A}(q, y)) = y_j$$
($\star$ $\mathcal{C}$ reconstructs the correct value of y_j for a random y $\star$)
4. If the two conditions are true

then output "ACCEPT",

otherwise output "REJECT".

Figure 4: Distinguishing algorithm DISTRIBUTIONAL-DIST.

We now state our first result about PIR protocols with reconstruction errors.

Theorem 5.5 *If there is a single-server PIR protocol in which the server communicates less than n bits and the reconstruction error is at most* $1/(8n)$ *then one-way functions exist.*

Proof: The proof is similar to the proof of Theorem 4.4. Let
$$f(i, r) \stackrel{\text{def}}{=} \langle i, \mathcal{Q}(1^n, i, r)\rangle.$$
Assume that f is not a distributional one-way function. Hence, there exists an algorithm D-INV such that the statistical distance between the distribution $\langle i, r, i, \mathcal{Q}(1^n, i, r)\rangle$ and the distribution $\langle \text{D-INV}(i, \mathcal{Q}(1^n, i, r)), i, \mathcal{Q}(1^n, i, r)\rangle$ is at most $1/(20n^2)$ for infinitely many values of n, where i and r are chosen uniformly. This implies that for *every* i:

$$\text{The distance between } \langle i, r, i, \mathcal{Q}(1^n, i, r)\rangle \text{ and } \langle \text{D-INV}(i, \mathcal{Q}(1^n, i, r)), i, \mathcal{Q}(1^n, i, r)\rangle \le \tfrac{1}{20n}, \quad (3)$$

where the random string r and the random choices of D-INV are distributed uniformly. In Figure 4 we show how to use D-INV to construct an algorithm that distinguishes between $\mathcal{Q}(1^n, i_n, \cdot)$ and $\mathcal{Q}(1^n, j_n, \cdot)$ for some i_n and j_n.

By the assumption that the PIR protocol has error at most $1/(8n)$, the probability that
$$\mathcal{C}(1^n, j, r, \mathcal{A}(\mathcal{Q}(1^n, j, r), y)) \neq y_j$$
is at most $1/(8n)$, where the probability is taken over the choice of r and y. Thus, by (3), if r is chosen at random and $q = \mathcal{Q}(1^n, j, r)$ then Algorithm DISTRIBUTIONAL-DIST outputs "REJECT" with probability at most $1/(8n)+1/(20n) < 1/(5n)$, where the probability is taken over r, y, and the random choices of D-INV.

We next claim that for every i there exists an index j such that if r is chosen at random and $q = \mathcal{Q}(1^n, i, r)$ then the probability that Algorithm DISTRIBUTIONAL-DIST outputs "REJECT" is at least $1/(4n)$, where the probability is taken over r, y, and the random choices of D-INV. Otherwise, execute Protocol TRANSMIT, described in Figure 3, where D-INV is used instead of INV. With probability at least $3/4$ the user $\mathcal{U}$ reconstructs the correct y in this protocol, where the probability is taken over r, y, and the random choices of D-INV (unlike the proof of Lemma 4.2 where the protocol succeeded for every y). This implies that we can fix r and the random choices of D-INV such that the protocol succeeds for a fraction of $3/4$ of the strings $y \in \{0,1\}^n$. But in any transmission protocol in which the deterministic user (receiver) can reconstruct the correct string for $(3/4)\cdot 2^n$ strings the server (sender) must send at least n bits. The assumption that the server sends less than n bits in the PIR protocol implies the existence of an index j as required.

To complete the proof we show, assuming that f is not a distributional one-way function, an algorithm that distinguishes between the distributions $\mathcal{Q}(1^n, i_n, \cdot)$ and $\mathcal{Q}(1^n, j_n, \cdot)$ for some indices i_n and j_n: Fix any i_n and let j_n be the index guaranteed in the previous paragraph. If we choose r uniformly, compute $q = \mathcal{Q}(1^n, j_n, r)$, and run Algorithm DISTRIBUTIONAL-DIST with input $\langle q, j_n\rangle$ then this algorithm outputs "REJECT" with probability at most $1/(5n)$. On the other hand, if we choose r uniformly, compute $q = \mathcal{Q}(1^n, i_n, r)$, and run Algorithm DISTRIBUTIONAL-DIST with input $\langle q, j_n\rangle$ then this algorithm outputs "REJECT" with probability at least $1/(4n)$. This contradicts the privacy requirement, and therefore the assumption that f is not a distributional one-way function is false. □

Theorem 5.5 allows only a polynomially small error in the reconstruction. We next prove an analogous theorem for protocols with a constant probability of error.

Corollary 5.6 *If there is a protocol with error* ϵ*, where* ϵ *is a constant and* $0 \le \epsilon < 0.5$*, and the server communicates less than* $(1 - H(\epsilon))\cdot n - 1$ *bits then one-way functions exist (where* $H(\epsilon)$ *is the binary entropy function; that is,* $H(\epsilon) = -\epsilon\log\epsilon - (1-\epsilon)\log(1-\epsilon)$*).*

Proof: The proof is similar to the proof of Theorem 5.5. In this case if r is chosen at random and $q = \mathcal{Q}(1^n, j, r)$ then the probability that Algorithm DISTRIBUTIONAL-DIST outputs "REJECT" is at most $\epsilon + 1/(20n)$, where the probability is taken over r, y, and the random choices of D-INV. Thus, to prove the corollary we only have to argue that for every i_n there exists a j_n such that if r is chosen at random and $q = \mathcal{Q}(1^n, i, r)$ then the probability that Algorithm DISTRIBUTIONAL-DIST outputs "REJECT" is at least $\epsilon + 1/n$, where the probability is taken over r, y, and the random choices of D-INV. If this is not the case then in Protocol TRANSMIT the server $\mathcal{S}$ sends a "short" message and the user $\mathcal{U}$ can reconstruct each bit correctly with probability at least $1 - \epsilon - 1/n$. In Lemma A.1, which appears in Appendix A, we prove that this is impossible. □

References

[1] M. Abadi, J. Feigenbaum, and J. Kilian. On hiding information from an oracle. *J. of Computer and System Sciences*, 39:21–50, 1989.

[2] A. Ambainis. Upper bound on the communication complexity of private information retrieval. In *Proc. of 24th ICALP*, volume 1256 of *Lecture Notes in Computer Science*, pages 401–407, 1997.

[3] D. Beaver and J. Feigenbaum. Hiding instances in multioracle queries. In C. Choffrut and T. Lengauer, editors, *STACS '90, 7th Annu. Symp. on Theoretical Aspects of Computer Science*, volume 415 of *Lecture Notes in Computer Science*, pages 37–48. Springer-Verlag, 1990.

[4] D. Beaver, J. Feigenbaum, J. Kilian, and P. Rogaway. Locally random reductions: Improvements and applications. *J. of Cryptology*, 10(1):17–36, 1997. Early version: Security with small communication overhead, CRYPTO '90, volume 537 of *Lecture Notes in Computer Science*, pages 62-76. Springer-Verlag, 1991.

[5] C. Cachin, S. Micali, and M. Stadler. Computationally private information retrieval with polylogarithmic communication. In *Advances in Cryptology - EUROCRYPT '99*, 1999. To appear.

[6] B. Chor and N. Gilboa. Computationally private information retrieval. In *Proc. of the 29th Annu. ACM Symp. on the Theory of Computing*, pages 304–313, 1997.

[7] B. Chor and O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM J. on Computing*, 17(2):230–261, 1988.

[8] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In *Proc. of the 36th Annu. IEEE Symp. on Foundations of Computer Science*, pages 41–51, 1995. Journal version to appear in JACM.

[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 C. Pomerance, editor, *Advances in Cryptology - CRYPTO '87*, volume 293 of *Lecture Notes in Computer Science*, pages 350–354. Springer-Verlag, 1988.

[11] G. Di Crescenzo, Y. Ishai, and R. Ostrovsky. Universal service-providers for database private information retrieval. In *Proc. of the 17th Annu. ACM Symp. on Principles of Distributed Computing*, pages 91–100, 1998.

[12] G. Di Crescenzo, T. Malkin, and R. Ostrovsky. Single-database private information retrieval implies oblivious transfer. Manuscript, November, 1998.

[13] Y. Gertner, S. Goldwasser, and T. Malkin. A random server model for private information retrieval. In M. Luby, J. Rolim, and M. Serna, editors, *RANDOM '98, 2nd International Workshop on Randomization and Approximation Techniques in Computer Science*, volume 1518 of *Lecture Notes in Computer Science*, pages 200–217. Springer, 1998.

[14] Y. Gertner, Y. Ishai, E. Kushilevitz, and T. Malkin. Protecting data privacy in private information retrieval schemes. In *Proc. of the 30th Annu. ACM Symp. on the Theory of Computing*, pages 151–160, 1998.

[15] O. Goldreich. Note on computational indistinguishability. *Inform. Process. Lett.*, 34(6):277–281, 1990.

[16] O. Goldreich. *Foundations of Cryptography (fragments of a book)*. Electronic Colloquium on Computational Complexity, 1995. Electronic publication: http://www.eccc.uni-trier.de/eccc-local/ECCC-Books/eccc-books.html.

[17] S. Goldwasser and S. Micali. Probabilistic encryption. *J. of Computer and System Sciences*, 28(21):270–299, 1984.

[18] J. Hastad, R. Impagliazzo, L. A. Levin, and M. Luby. Construction of a pseudo-random generator from any one-way function. Technical Report TR-91-068, International Computer Science Institute, 1991.

[19] R. Impagliazzo and M. Luby. One-way functions are essential for complexity based cryptography. In *Proc. of the 30th Annu. IEEE Symp. on Foundations of Computer Science*, pages 230–235, 1989.

[20] R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *Proc. of the 21st Annu. ACM Symp. on the Theory of Computing*, pages 44–61, 1989.

[21] Y. Ishai and E. Kushilevitz. Improved upper bounds on information theoretic private information retrieval. In *Proc. of the 31th Annu. ACM Symp. on the Theory of Computing*, 1999.

[22] E. Kushilevitz and N. Nisan. *Communication Complexity.* Cambridge University Press, 1997.

[23] E. Kushilevitz and R. Ostrovsky. Replication is not needed: Single database, computationally-private information retrieval. In *Proc. of the 38th Annu. IEEE Symp. on Foundations of Computer Science*, pages 364–373, 1997.

[24] M. Luby. *Pseudorandomness and Cryptographic Applications.* Princeton University Press, 1996.

[25] E. Mann. Private access to distributed information. Master's thesis, Technion - Israel Institute of Technology, Haifa, 1998.

[26] M. Naor. Bit commitment using pseudorandom generators. *J. of Cryptology*, 4:151–158, 1991.

[27] R. Ostrovsky and V. Shoup. Private information storage. In *Proc. of the 29th Annu. ACM Symp. on the Theory of Computing*, pages 294–303, 1997.

[28] R. Ostrovsky and A. Wigderson. One-way functions are essential for non-trivial zero-knowledge. In *2nd Israel Symp. on Theory of Computing and Systems*, pages 3–17, 1993.

[29] M. O. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Harvard Aiken Computation Laboratory, 1981.

[30] J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *Proc. of the 22nd Annu. ACM Symp. on the Theory of Computing*, pages 387–394, 1990.

[31] A. C. Yao. Theory and application of trapdoor functions. In *Proc. of the 23th Annu. IEEE Symp. on Foundations of Computer Science*, pages 80–91, 1982.

A Lower Bound for Transmitting a Database

The next lemma shows (using standard information-theory arguments) that if there is a protocol between a server and a user such that for every index i the user can reconstruct y_i with some constant probability greater than 1/2 then the number of bits sent by the server is $\Omega(n)$. We assume that the reader is familiar with the notions of entropy and mutual information. For background on information-theory see, e.g., [9].

Lemma A.1 *Consider a server $\mathcal{S}$ holding an n bit string y, and let $\epsilon < 0.5$ be a constant. Assume there is a (possibly multi-round) protocol between $\mathcal{S}$ and a user in which the output of the user is a string y' such that $\Pr[y_i = y'_i] \geq 1 - \epsilon$ for every $i \in \{1, \ldots, n\}$, where the probability is over the uniform distribution of $y \in \{0,1\}^n$ and the random choices of the user and the server. Then, the number of bits sent by $\mathcal{S}$ is at least $(1 - H(\epsilon)) \cdot n$.*

Proof: Notice that the user has no input in this protocol and that we do not count the number of bits he sends. However, the user might use randomization. Nevertheless, if the user sends its random string to $\mathcal{S}$ in the first step of the protocol then the server can simulate the rest of the protocol without interacting with the user. The server then sends its part of the simulated protocol to the user, who reconstructs y' as before. Furthermore, we can assume that $\mathcal{S}$ is deterministic since it can get random coins from the user. So, without loss of generality, we can assume that the protocol has the following very simple structure: first, the user picks a random string and sends it to the server; then, a deterministic $\mathcal{S}$ sends its reply, and finally the user computes y' from the random string and the reply.

Next, we introduce some notation. Let R, A, Y, Y' be random variables, where R represents the message sent by the user, A represents the answer sent by the server to the user, Y represents the string held by the server, and Y' represents the string reconstructed by the user. Finally, Y_i and Y'_i represent the i-th bit of Y and Y' respectively.

The bit Y'_i is an estimation of Y_i obtained from R and A. Thus, by Fano Inequality for binary random variables,

$$H(Y_i|RA) \leq H(p_E), \quad \text{where } p_E \stackrel{\text{def}}{=} \Pr[Y_i \neq Y'_i].$$

By the assumptions of the theorem $p_E \leq \epsilon < 0.5$, thus $H(Y_i|RA) \leq H(\epsilon)$. By the chain rule for the entropy

$$\begin{aligned} H(Y|RA) &= \sum_{k=1}^{n} H(Y_k|Y_1 \ldots Y_{k-1}RA) \\ &\leq \sum_{k=1}^{n} H(Y_k|RA) \;\leq\; H(\epsilon) \cdot n. \end{aligned}$$

Recall that Y and R are independent random variables and Y is uniformly distributed in $\{0,1\}^n$, therefore

$$H(Y|R) = H(Y) = n.$$

Combining the last two inequalities we get

$$\begin{aligned} I(Y;A|R) &= H(Y|R) - H(Y|RA) \\ &\geq (1 - H(\epsilon)) \cdot n. \end{aligned}$$

Thus,

$$\begin{aligned} (1 - H(\epsilon)) \cdot n &\leq I(Y;A|R) \;\leq\; H(A|R) \\ &\leq H(A) \;\leq\; \log|A|, \end{aligned}$$

where $|A|$ is the size of the domain of the random variable A, and thus $\log|A|$ is a lower bound on the number of bits that the server needs to send to the user. $\square$