Exponential Lower Bound for 2-Query Locally Decodable Codes via a Quantum Argument*

Iordanis Kerenidis†
UC Berkeley, CS Division
587 Soda Hall
Berkeley, CA 94720
U.S.A.
jkeren@cs.berkeley.edu

Ronald de Wolf‡
CWI, INS4
Kruislaan 413
1098 SJ Amsterdam
The Netherlands
rdewolf@cwi.nl

June 17, 2004

Abstract

A *locally decodable code* encodes n-bit strings x in m-bit codewords $C(x)$ in such a way that one can recover any bit x_i from a corrupted codeword by querying only a few bits of that word. We use a *quantum* argument to prove that LDCs with 2 classical queries require exponential length: $m = 2^{\Omega(n)}$. Previously this was known only for linear codes (Goldreich et al. 02). The proof proceeds by showing that a 2-query LDC can be decoded with a single quantum query, when defined in an appropriate sense. It goes on to establish an exponential lower bound on any '1-query locally quantum-decodable code'. We extend our lower bounds to non-binary alphabets and also somewhat improve the polynomial lower bounds by Katz and Trevisan for LDCs with more than 2 queries. Furthermore, we show that q quantum queries allow more succinct LDCs than the best known LDCs with q classical queries. Finally, we give new classical lower bounds and quantum upper bounds for the setting of private information retrieval. In particular, we exhibit a quantum 2-server PIR scheme with $O(n^{3/10})$ qubits of communication, beating the $O(n^{1/3})$ bits of communication of the best known classical 2-server PIR.

Keywords: Quantum computing, locally decodable codes, private information retrieval.

1 Introduction

1.1 Setting

Error correcting codes allow one to encode an n-bit string x into an m-bit codeword $C(x)$ in such a way that x can still be recovered even if the codeword is corrupted in a number of places. For example, codewords of length $m = O(n)$ already suffice to recover from errors in a constant fraction of the bitpositions of the codeword, even in linear time [33]. One disadvantage of such "standard" error correction is that one usually needs to consider all or most of the (corrupted)

*A preliminary version of this paper appeared in STOC'03 [22].
†Supported by DARPA under agreement number F 30602–01-2–0524. Part of this work was done when visiting CWI.
‡Most of this work was done while a postdoc at UC Berkeley, supported by Talent grant S 62–565 from the Netherlands Organization for Scientific Research (NWO). Also (partially) funded by projects QAIP (IST–1999–11234) and RESQ (IST-2001-37559) of the IST-FET programme of the EC.

codeword to recover anything about x. If one is only interested in recovering one or a few of the bits of x, then more efficient schemes are possible, which allow us to extract small parts of encoded information from a corrupted codeword, while looking at ("querying") only a few positions of that word. Such schemes are called locally decodable codes (LDCs). They have found various applications in complexity theory and cryptography, such as self-correcting computations [5, 24, 17, 16, 18], Probabilistically Checkable Proofs [2], worst-case to average-case reductions [3, 34], private information retrieval [11], and extractors [25]. Informally, LDCs are described as follows:

> A (q, δ, ε)-*locally decodable code* encodes n-bit strings x into m-bit codewords $C(x)$, such that, for each i, the bit x_i can be recovered with probability $1/2 + \varepsilon$ making only q queries, even if the codeword is corrupted in δm of the bits.

For example, the Hadamard code is a locally decodable code where *two* queries suffice for predicting any bit with constant advantage, even with a constant fraction of errors. The code has $m = 2^n$ and $C(x)_j = j \cdot x \bmod 2$ for all $j \in \{0,1\}^n$. Recovery from a corrupted codeword y is possible by picking a random $j \in \{0,1\}^n$, querying y_j and $y_{j \oplus e_i}$, and computing the XOR of those two bits as our guess for x_i. If neither of the two queried bits has been corrupted, then we output $y_j \oplus y_{j \oplus e_i} = j \cdot x \oplus (j \oplus e_i) \cdot x = e_i \cdot x = x_i$, as we should. If $C(x)$ has been corrupted in at most δm positions, then a fraction of at least $1 - 2\delta$ of all $(j, j \oplus e_i)$ pairs of indices is uncorrupted, so the recovery probability is at least $1 - 2\delta$. This is $> 1/2$ as long as $\delta < 1/4$. The main drawback of the Hadamard code is its exponential length.

Clearly, we would like both the codeword length m and the number of queries q to be small. The main complexity question about LDCs is how large m needs to be, as a function of n, q, δ, and ε. For $q = polylog(n)$, Babai et al. [2] showed how to achieve almost linear size codes, for some fixed δ and ε. Beimel et al. [8] recently improved the best known upper bounds for constant q to $m = 2^{n^{O(\log\log q / q \log q)}}$, with some more precise bounds for small q.

The study of *lower* bounds on m was initiated by Katz and Trevisan [21]. They proved that for $q = 1$, LDCs do not exist if n is larger than some constant depending on δ and ε. For $q \geq 2$, they proved a bound of $m = \Omega(n^{1+1/(q-1)})$ if the q queries are made non-adaptively; this bound was generalized to the adaptive case by Deshpande et al. [14]. This establishes superlinear (but at most quadratic) lower bounds on the length of LDCs with a constant number of queries. There is still a large gap between the best known upper and lower bounds. In particular, it is open whether $m = poly(n)$ is achievable with constant q. Goldreich et al. [20] examined the case $q = 2$, and showed that $m \geq 2^{\delta\varepsilon n/8}$ if C is a *linear* code. Obata [29] subsequently strengthened the dependence on ε to $m \geq 2^{\Omega(\delta n/(1-2\varepsilon))}$, which is essentially optimal. Very recently, Ben-Sasson et al. [9] studied a *relaxed* notion of LDCs where the decoder is allowed to output "don't know" for a constant fraction of the indices. They construct relaxed LDCs with a constant number of queries and size $m = n^{1+\epsilon}$.

Katz and Trevisan, and Goldreich et al. established a close connection between locally decodable codes and *private information retrieval (PIR)* schemes. A PIR scheme allows a user to extract a bit x_i with probability $1/2 + \varepsilon$ from an n-bit database x that is replicated over some $k \geq 1$ servers, without the server(s) learning *which* i the user wants. The main complexity measure of a PIR scheme is its communication complexity, i.e., the sum of the lengths of the queries that the user sends to each server, and the length of the servers' answers. Roughly, the queries in an LDC correspond to the servers in a PIR scheme. In fact, the best known LDCs for constant q are derived from PIR schemes.

If there is only one server ($k = 1$), then privacy can be maintained by letting the server send the whole n-bit database to the user. This takes n bits of communication and is optimal. If the database is replicated over $k \geq 2$ servers, then there exist protocols with significantly less communication. Chor et al. [11] exhibited a 2-server PIR scheme with communication complexity $O(n^{1/3})$ and one with $O(n^{1/k})$ for $k > 2$. Ambainis [1] improved the latter to $O(n^{1/(2k-1)})$. Beimel et al. [8] improved the communication complexity to $O(n^{2\log\log k/k\log k})$. Their results improve the previous best bounds for all $k \geq 3$ but not for $k = 2$.

No general lower bounds better than $\Omega(\log n)$ are known for PIRs with $k \geq 2$ servers. For the case of 2 servers, the best known lower bound is $4\log n$, due to Mann [26]. A PIR scheme is *linear* if for every query that the user makes, the answer bits are *linear combinations* of the bits of x. Goldreich et al. [20] proved that linear 2-server PIRs with t-bit queries and a-bit answers where the user looks only at k predetermined positions in each answer require $t = \Omega(n/a^k)$.

1.2 Results: Locally Decodable Codes

The main result of this paper is an exponential lower bound for general 2-query LDCs:

Theorem 4 If $C : \{0,1\}^n \to \{0,1\}^m$ is a $(2, \delta, \varepsilon)$-locally decodable code, then

$$m \geq 2^{cn-1},$$

for $c = 3\delta\varepsilon^2/(98\ln 2)$.

This is the first superpolynomial lower bound on general LDCs with more than one query. Our constant c in the exponent is somewhat worse than those of Goldreich et al. [20] and of Obata [29], but our proof establishes the exponential lower bound for *all* LDCs, not just linear ones.

Our proof introduces one radically new ingredient: *quantum* computing. We show that if two classical queries can recover x_i with probability $1/2 + \varepsilon$, then x_i can also be recovered with probability $1/2 + 4\varepsilon/7$ using only one "quantum query". In other words, a $(2, \delta, \varepsilon)$-locally decodable code is a $(1, \delta, 4\varepsilon/7)$-locally *quantum*-decodable code. We then prove an exponential lower bound for 1-query LQDCs by showing, roughly speaking, that a 1-query LQDC of length m induces a *quantum random access code* for x of length about $\log m$. Such a code enables its user to recover each bit x_i of his choice. Nayak's [27] linear lower bound on the length of such codes finishes off the proof. For the sake of completeness, we include a proof of his result in Appendix B.

This lower bound for classical LDCs is one of the very few examples where tools from quantum computing enable one to prove *new* results in *classical* computer science. We know only a few other examples of this.[1] Radhakrishnan et al. [30] proved lower bounds for the set membership data structure that hold for quantum algorithms, but are in fact stronger than the previous classical lower bounds of Buhrman et al. [10]. Sen and Venkatesh did the same for data structures for the predecessor problem [32, quant-ph version]. Klauck et al. [23] proved lower bounds for the k-round quantum communication complexity of the tree-jumping problem that are somewhat stronger than the previous best classical lower bounds. In cryptography, Gisin, Renner, and Wolf [19] used an analogy with "quantum bound entanglement" to provide evidence against the conjecture that the "intrinsic information" in a random variable shared by Alice, Bob, and eavesdropper Eve always equals the amount of secret key that Alice and Bob can extract from this; later this conjecture was indeed disproved [31], though without using quantum methods. In all these cases, the underlying proof techniques easily yield a classical proof: one just replaces quantum notions like von Neumann entropy and trace distance by their classical analogues to get a classical proof for the classical

[1] The quantum lower bound on the communication complexity of the inner product function of Cleve et al. [12] provides new insight in a classical result, but does not establish a *new* result for classical computer science.

case. In contrast, our proof seems to be more "inherently quantum" since there is no classical analog of our 2-classical-queries-to-1-quantum-query reduction (2-query LDCs exist but 1-query LDCs don't).

While Section 3 focuses only on codes over the *binary* alphabet, in Section 4.1 we extend our result to the case of larger alphabets, using a classical reduction due to Trevisan [35]. In Section 4.2 we look at LDCs with $q \geq 3$ queries and improve the polynomial lower bounds of Katz and Trevisan [21]. Our bounds are still polynomial and far from the best known upper bounds. In Section 4.3 we observe that our construction implies the existence of 1-query quantum-decodable codes for all n. The Hadamard code is an example of this. Here the codewords are still classical, but the decoder is quantum. As mentioned before, if we only allow one *classical* query, then LDCs do not exist for n larger than some constant depending on δ and ε [21]. For larger q, it turns out that the best known $(2q, \delta, \varepsilon)$-LDCs, due to Beimel et al. [8], are actually (q, δ, ε)-LQDCs. Hence for fixed number of queries q, we obtain LQDCs that are significantly shorter than the best known LDCs. In particular, Beimel et al. give a 4-query LDC with length $m = 2^{O(n^{3/10})}$ which is a 2-query LQDC. This is significantly shorter than the $m = 2^{\Theta(n)}$ that 2-query LDCs need. We summarize the situation in Table 1, where our contributions are indicated by boldface.

Queries	Length of LDC	Length of LQDC
$q = 1$	don't exist	$\mathbf{2^{\Theta(n)}}$
$q = 2$	$\mathbf{2^{\Theta(n)}}$	$\mathbf{2^{O(n^{3/10})}}$
$q = 3$	$2^{O(n^{1/2})}$	$\mathbf{2^{O(n^{1/7})}}$
$q = 4$	$2^{O(n^{3/10})}$	$\mathbf{2^{O(n^{1/11})}}$

Table 1: Best known bounds on the length of LDCs and LQDCs with q queries

1.3 Results: Private Information Retrieval

In the private information retrieval setting, our techniques allow us to reduce classical 2-server PIR schemes with 1-bit answers to quantum 1-server PIRs, which in turn can be reduced to a random access code [27]. Thus in Section 5.1 we obtain an $\Omega(n)$ lower bound on the communication complexity for all classical 2-server PIRs with 1-bit answers. In Section 5.2 we extend our lower bound to PIR schemes with larger answers. Previously, such a bound was known only for *linear* PIRs (first proven in [11, Section 5.2] for 1-bit answers and extended to constant-length answers in [20]). Furthermore, our results combined with those of Katz and Trevisan give a $4.4 \log n$ lower bound for the general 2-server PIR. This is the first, very modest improvement on the bound of Mann [26]. Subsequently to our work, Beigel, Fortnow, and Gasarch [7] found a classical proof that a 2-server PIR with *perfect* recovery ($\varepsilon = 1/2$) and 1-bit answers needs query length $\geq n - 2$. However, their proof does not seem to extend to the case $\varepsilon < 1/2$, or to larger answers.

Apart from giving new lower bounds for *classical* PIR, we can also use our 2-to-1 reduction to obtain *quantum* PIR schemes that beat the best known classical PIRs. In particular, Beimel et al. [8, Example 4.2] exhibit a classical 4-server PIR scheme with 1-bit answers and communication complexity $O(n^{3/10})$. We can reduce this to a quantum 2-server PIR with $O(n^{3/10})$ qubits of communication. This beats the best known classical 2-server PIR, which has complexity $O(n^{1/3})$. We can similarly give quantum improvements over the best known k-server PIR schemes for $k > 2$. However, this does not constitute a true classical-quantum separation in the PIR setting yet, since no good lower bounds are known for classical PIR. We summarize the best known bounds for classical and quantum PIR in Table 2.

Servers	PIR complexity	QPIR complexity
$k=1$	$\Theta(n)$	$\Theta(n)$
$k=2$	$O(n^{1/3})$	$\boldsymbol{O(n^{3/10})}$
$k=3$	$O(n^{1/5.25})$	$\boldsymbol{O(n^{1/7})}$
$k=4$	$O(n^{1/7.87})$	$\boldsymbol{O(n^{1/11})}$

Table 2: Best known bounds on the communication complexity of classical and quantum PIR

2 Preliminaries

2.1 Quantum

Below we give more precise definitions of locally decodable codes, PIR schemes, and related notions, but we first explain the standard notation of quantum computing.

Let H denote a 2-dimensional complex vector space, equipped with the standard inner product. We pick an orthonormal basis for this space, label the two basis vectors $|0\rangle$ and $|1\rangle$, and for simplicity identify them with the vectors $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$, respectively. A *qubit* is a unit length vector in this space, and so can be expressed as a linear combination of the basis states:

$$\alpha_0|0\rangle + \alpha_1|1\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix}.$$

Here α_0, α_1 are complex *amplitudes*, and $|\alpha_0|^2 + |\alpha_1|^2 = 1$.

An *m-qubit system* is a unit vector in the m-fold tensor space $H \otimes \cdots \otimes H$. The 2^m basis states of this space are the m-fold tensor products of the states $|0\rangle$ and $|1\rangle$. For example, the basis states of a 2-qubit system are the four 4-dimensional unit vectors $|0\rangle \otimes |0\rangle$, $|0\rangle \otimes |1\rangle$, $|1\rangle \otimes |0\rangle$, and $|1\rangle \otimes |1\rangle$. We abbreviate, e.g., $|1\rangle \otimes |0\rangle$ to $|0\rangle|1\rangle$, or $|1,0\rangle$, or $|10\rangle$, or even $|2\rangle$ (since 2 is 10 in binary). With these basis states, an m-qubit state $|\phi\rangle$ is a 2^m-dimensional complex unit vector

$$|\phi\rangle = \sum_{i\in\{0,1\}^m} \alpha_i|i\rangle.$$

We use $\langle\phi| = |\phi\rangle^*$ to denote the conjugate transpose of the vector $|\phi\rangle$, and $\langle\phi|\psi\rangle = \langle\phi|\cdot|\psi\rangle$ for the inner product between states $|\phi\rangle$ and $|\psi\rangle$. These two states are *orthogonal* if $\langle\phi|\psi\rangle = 0$. The *norm* of $|\phi\rangle$ is $\parallel \phi \parallel = \sqrt{\langle\phi|\phi\rangle}$.

A *mixed state* $\{p_i, |\phi_i\rangle\}$ is a classical distribution over pure quantum states, where the system is in state $|\phi_i\rangle$ with probability p_i. We can represent a mixed quantum state by the *density matrix* which is defined as $\rho = \sum_i p_i|\phi_i\rangle\langle\phi_i|$. Note that ρ is a positive semidefinite operator with trace (sum of diagonal entries) equal to 1. The density matrix of a pure state $|\phi\rangle$ is $\rho = |\phi\rangle\langle\phi|$.

A quantum system is called *bipartite* if it consists of two subsystems. We can describe the state of each of these subsystems separately with the *reduced density matrix*. For example, if a quantum state has the form $|\phi\rangle = \sum_i \sqrt{p_i}|i\rangle|\phi_i\rangle$, then the state of a system holding only the second part of $|\phi\rangle$ is described by the (reduced) density matrix $\sum_i p_i|\phi_i\rangle\langle\phi_i|$.

A quantum state can evolve by a unitary operation or by a measurement. A *unitary* transformation is a linear mapping that preserves the ℓ_2 norm. If we apply a unitary U to a state $|\phi\rangle$, it evolves to $U|\phi\rangle$. A mixed state ρ evolves to $U\rho U^\dagger$.

The most general measurement allowed by quantum mechanics is specified by a family of positive semidefinite operators $E_i = M_i^* M_i$, $1 \le i \le k$, subject to the condition that $\sum_i E_i = I$. Given a density matrix ρ, the probability of observing the ith outcome under this measurement is given by the trace $p_i = \mathrm{Tr}(E_i\rho) = \mathrm{Tr}(M_i\rho M_i^*)$. These p_i are nonnegative because E_i and ρ are positive semidefinite. They also sum to 1, as they should:

$$\sum_{i=1}^{k} p_i = \sum_{i=1}^{k} \mathrm{Tr}(E_i\rho) = \mathrm{Tr}(\sum_{i=1}^{k} E_i\rho) = \mathrm{Tr}(I\rho) = 1.$$

If the measurement yields outcome i, then the resulting quantum state is $M_i\rho M_i^*/\mathrm{Tr}(M_i\rho M_i^*)$. In particular, if $\rho = |\phi\rangle\langle\phi|$, then $p_i = \langle\phi|E_i|\phi\rangle = \| M_i|\phi\rangle \|^2$, and the resulting state is $M_i|\phi\rangle/\| M_i|\phi\rangle \|$. A special case is where $k = 2^m$ and $B = \{|\psi_i\rangle\}$ forms an orthonormal basis of the m-qubit space. "Measuring in the B-basis" means that we apply the measurement given by $E_i = M_i = |\psi_i\rangle\langle\psi_i|$. Applying this to a pure state $|\phi\rangle$ gives resulting state $|\psi_i\rangle$ with probability $p_i = |\langle\phi|\psi_i\rangle|^2$.

Finally, a word about quantum *queries*. A query to an m-bit string y is commonly formalized as the following unitary transformation, where $j \in [m]$, and $b \in \{0,1\}$ is called the *target bit*:

$$|j\rangle|b\rangle \mapsto |j\rangle|b \oplus y_j\rangle.$$

A quantum computer may apply this to any superposition. An equivalent formalization that we will be using here, is:

$$|c\rangle|j\rangle \mapsto (-1)^{c\cdot y_j}|c\rangle|j\rangle.$$

Here c is a *control bit* that controls whether the phase $(-1)^{y_j}$ is added or not. Given some extra workspace, one query of either type can be simulated exactly by one query of the other type.

We refer to Nielsen and Chuang [28] for more details.

2.2 Codes

Below, by a 'decoding algorithm' we mean an algorithm (quantum or classical depending on context) with oracle access to the bits of some (possibly corrupted) codeword y for x. The algorithm gets input i and is supposed to recover x_i, making only few queries to y. We want to emphasize that we speak of an 'algorithm' merely as a convenient way to formalize the decoding process. Our focus is not the algorithmics of the decoding but its information-theoretic aspects, i.e., the tradeoff between the number q of queries allowed for decoding and the required codelength m.

Definition 1 *$C : \{0,1\}^n \to \{0,1\}^m$ is a (q, δ, ε)-*locally decodable code *(LDC) if there is a classical randomized decoding algorithm A such that*

1. *A makes at most q queries to m-bit string y, non-adaptively.*
2. *For all x and i, and all $y \in \{0,1\}^m$ with Hamming distance $d(C(x), y) \leq \delta m$ we have $\Pr[A^y(i) = x_i] \geq 1/2 + \varepsilon$.*

The LDC is called linear *if C is a linear function over $GF(2)$ (i.e., $C(x + y) = C(x) + C(y)$).*

By allowing A to be a quantum computer and to make queries in superposition, we can similarly define (q, δ, ε)-locally quantum-*decodable codes (LQDCs).*

It will be convenient to work with *non-adaptive* queries, as used in the above definition, so the distribution on the queries that A makes is independent of y. However, our main lower bound also holds for adaptive queries, see the first remark at the end of Section 3.3.

2.3 Private Information Retrieval

Next we define private information retrieval schemes.

Definition 2 *A one-round, $(1-\delta)$-secure, k-server private information retrieval (PIR) scheme with recovery probability $1/2+\varepsilon$, query size t, and answer size a, consists of a randomized algorithm (the user), and k deterministic algorithms $S_1, \ldots, S_k$ (the servers), such that*

1. *On input $i \in [n]$, the user produces k t-bit queries $q_1, \ldots, q_k$ and sends these to the respective servers. The jth server sends back an a-bit string $a_j = S_j(x, q_j)$. The user outputs a bit b depending on $i, a_1, \ldots, a_k$, and his randomness.*
2. *For all x and i, the probability (over the user's randomness) that $b = x_i$ is at least $1/2+\varepsilon$.*
3. *For all x and j, the distributions on q_j (over the user's randomness) are δ-close (in total variation distance) for different i.*

The scheme is called linear *if, for every j and q_j, the jth server's answer $S_j(x, q_j)$ is a linear combination over $GF(2)$ of the bits of x.*

We can straightforwardly generalize these definitions to quantum *PIR for the case where $\delta = 0$ (the server's state after the query should be independent of i). That is the only case we need here.*

All known upper bounds on PIR have one round of communication, $\varepsilon = 1/2$ (perfect recovery) and $\delta = 0$ (the servers get no information whatsoever about i). Below we will assume one round and $\delta = 0$ without mentioning this further.

3 Lower Bound for 2-Query Locally Decodable Codes

Our proof has two parts, each with a clear intuition but requiring quite a few technicalities:

1. A 2-query LDC is a 1-query LQDC, because one quantum query can compute the same Boolean functions as two classical queries (albeit with slightly worse error probability).
2. The length m of a 1-query LQDC must be exponential, because a uniform superposition over all indices contains only $\log m$ qubits, but induces a *quantum random access code* for x, for which a linear lower bound is already known [27].

3.1 From 2 Classical to 1 Quantum Query

The key to the first step is the following lemma:

Lemma 1 *Let $f : \{0,1\}^2 \to \{0,1\}$ and suppose we can make queries to the bits of some input string $a = a_1 a_2 \in \{0,1\}^2$. There exists a quantum algorithm that makes only one query (one that is independent of f) and outputs $f(a)$ with probability* exactly $11/14$, *and outputs $1 - f(a)$ otherwise.*

Proof. If we could construct the state

$$|\psi_a\rangle = \frac{1}{2}(|0\rangle|1\rangle + (-1)^{a_1}|1\rangle|1\rangle + (-1)^{a_2}|1\rangle|2\rangle + (-1)^{a_1+a_2}|0\rangle|2\rangle)$$

with one quantum query then we could determine a with certainty, since the four possible states $|\psi_b\rangle$ ($b \in \{0,1\}^2$) form an orthonormal basis. We could also see these states as the Hadamard encoding of the strings $b \in \{0,1\}^2$. Unfortunately we cannot construct $|\psi_a\rangle$ perfectly with one query. Instead, we approximate this state by making the query

$$\frac{1}{\sqrt{3}}\left(|0\rangle|1\rangle + |1\rangle|1\rangle + |1\rangle|2\rangle\right),$$

where the first bit is the control bit, and the appropriate phase $(-1)^{a_j}$ is put in front of $|j\rangle$ if the control bit is 1. The result of the query is the state

$$|\phi\rangle = \frac{1}{\sqrt{3}}\left(|0\rangle|1\rangle + (-1)^{a_1}|1\rangle|1\rangle + (-1)^{a_2}|1\rangle|2\rangle\right).$$

The algorithm then measures this state $|\phi\rangle$ in the orthonormal basis consisting of the four states $|\psi_b\rangle$. The probability of getting outcome a is $|\langle\phi|\psi_a\rangle|^2 = 3/4$, and each of the other 3 outcomes has probability $1/12$. The algorithm now determines its output based on f and on the measurement outcome b. We distinguish 3 cases for f:

1. $|f(1)^{-1}| = 1$ (the case $|f(1)^{-1}| = 3$ is completely analogous, with 0 and 1 reversed). If $f(b) = 1$, then the algorithm outputs 1 with probability 1. If $f(b) = 0$ then it outputs 0 with probability $6/7$ and 1 with probability $1/7$. Accordingly, if $f(a) = 1$, then the probability of output 1 is $\Pr[f(b) = 1] \cdot 1 + \Pr[f(b) = 0] \cdot 1/7 = 3/4 + 1/28 = 11/14$. If $f(a) = 0$, then the probability of output 0 is $\Pr[f(b) = 0] \cdot 6/7 = (11/12) \cdot (6/7) = 11/14$.

2. $|f(1)^{-1}| = 2$. Then $\Pr[f(a) = f(b)] = 3/4 + 1/12 = 5/6$. If the algorithm outputs $f(b)$ with probability $13/14$ and outputs $1 - f(b)$ with probability $1/14$, then its probability of output $f(a)$ is exactly $11/14$.

3. f is constant. In that case the algorithm just outputs that value with probability $11/14$.

Thus we always output $f(a)$ with probability $11/14$. □

Peter Høyer (personal communication) recently improved the 11/14 in the lemma to 9/10. We describe his algorithm in Appendix A and show that this success probability is best possible if we have only one quantum query.

Using our lemma we can prove:

Theorem 1 *A $(2,\delta,\varepsilon)$-LDC is a $(1,\delta,4\varepsilon/7)$-LQDC.*

Proof. Consider i, x, and y such that $d(C(x), y) \leq \delta m$. The 1-query quantum decoder will use the same randomness as the 2-query classical decoder. The random string of the classical decoder determines two indices $j, k \in [m]$ and an $f : \{0,1\}^2 \to \{0,1\}$ such that

$$\Pr[f(y_j, y_k) = x_i] = p \geq 1/2 + \varepsilon,$$

where the probability is taken over the decoder's randomness. We now use Lemma 1 to obtain a 1-query quantum decoder that outputs some bit b such that

$$\Pr[b = f(y_j, y_k)] = 11/14.$$

The success probability of this quantum decoder is:[2]

$$\begin{aligned}
\Pr[b = x_i] &= \Pr[b = f(y_j, y_k)] \cdot \Pr[f(y_j, y_k) = x_i] + \\
&\quad \Pr[b \neq f(y_j, y_k)] \cdot \Pr[f(y_j, y_k) \neq x_i] \\
&= \frac{11}{14}p + \frac{3}{14}(1-p) \\
&= \frac{3}{14} + \frac{4}{7}p \\
&\geq \frac{1}{2} + \frac{4\varepsilon}{7},
\end{aligned}$$

as promised. □

3.2 Lower Bound for 1-Query LQDCs

A quantum *random access code* is an encoding $x \mapsto \rho_x$ of n-bit strings x into m-qubit states ρ_x, possibly mixed, such that any bit x_i can be recovered with some probability $p \geq 1/2 + \varepsilon$ from ρ_x. The following lower bound is known on the length of such quantum codes [27] (see Appendix B).

Theorem 2 (Nayak) *An encoding $x \mapsto \rho_x$ of n-bit strings into m-qubit states with recovery probability at least p, has $m \geq (1 - H(p))n$.*

This allows us to prove an exponential lower bound for 1-query LQDCs:

Theorem 3 *If $C : \{0,1\}^n \to \{0,1\}^m$ is a $(1, \delta, \varepsilon)$-LQDC, then*

$$m \geq 2^{cn-1},$$

for $c = \delta\varepsilon^2/(16 \ln 2)$.

Proof. Our goal below is to show that we can recover each x_i with good probability from a number of copies of the uniform $\log(m) + 1$-qubit state

$$|U(x)\rangle = \frac{1}{\sqrt{2m}} \sum_{c \in \{0,1\}, j \in [m]} (-1)^{c \cdot C(x)_j} |c\rangle |j\rangle.$$

The intuitive reason for this is as follows. Since C is an LQDC, it is able to recover x_i even from a codeword that is corrupted in many (up to δm) places. Therefore the "distribution" of queries of the decoder must be "smooth", i.e., spread out over almost all the positions of the codeword—otherwise an adversary could choose the corrupted bits in a way that makes the recovery probability too low. The uniform distribution provides a reasonable approximation to such a "smooth" distribution. Since the uniform state $|U(x)\rangle$ is independent of i, we can actually recover *any* bit x_i with good probability, so it constitutes a quantum random access code for x. Applying Theorem 2 then gives the result.

[2] Here we use the 'exactly' part of Lemma 1. To see what could go wrong if the 'exactly' were 'at least', suppose the classical decoder outputs $\mathrm{AND}(y_1, y_2) = x_i$ with probability $3/5$ and $\mathrm{XOR}(y_3, y_4) = 1 - x_i$ with probability $2/5$. Then it outputs x_i with probability $3/5 > 1/2$. However, if our quantum procedure computes $\mathrm{AND}(y_1, y_2)$ with success probability $11/14$ but $\mathrm{XOR}(y_3, y_4)$ with success probability 1, then its recovery probability is $(3/5)(11/14) < 1/2$.

Let us be more precise. The most general query that the quantum decoder could make to recover x_i, is of the form

$$|Q_i\rangle = \sum_{c\in\{0,1\},j\in[m]} \alpha_{cj}|c\rangle|j\rangle|\phi_{cj}\rangle,$$

where the $|\phi_{cj}\rangle$ are pure states in the decoder's workspace and the α_{cj} are non-negative reals (any phases could be put in the $|\phi_{cj}\rangle$). This workspace can also incorporate any classical randomness used. However, the decoder could equivalently add these workspace states *after* the query, using the unitary map $|c\rangle|j\rangle|0\rangle \mapsto |c\rangle|j\rangle|\phi_{cj}\rangle$. Hence we can assume without loss of generality that the actual query is

$$|Q_i\rangle = \sum_{c\in\{0,1\},j\in[m]} \alpha_{cj}|c\rangle|j\rangle,$$

and that the decoder just measures the state resulting from this query. Let D and $I-D$ be the two measurement operators that the decoder uses for this measurement, corresponding to outputs 1 and 0, respectively. Its probability of giving output 1 on query-result $|R\rangle$ is $p(R) = \langle R|D|R\rangle$ (for clarity we don't write the $|\cdot\rangle$ inside the $p(\cdot)$).

Inspired by the smoothing technique of [21], we split the amplitudes α_j of the query $|Q_i\rangle$ into small and large ones: $A = \{(c,j) : \alpha_{cj} \leq \sqrt{1/\delta m}\}$ and $B = \{(c,j) : \alpha_{cj} > \sqrt{1/\delta m}\}$. Since the query does not affect the $|0\rangle|j\rangle$-states, we can assume without loss of generality that α_{0j} is the same for all j, so $\alpha_{0j} \leq 1/\sqrt{m} \leq 1/\sqrt{\delta m}$ and hence $(0,j) \in A$. Let $a = \sqrt{\sum_{(c,j)\in A} \alpha_{cj}^2}$ be the norm of the "small-amplitude" part. Since $\sum_{(c,j)\in B} \alpha_{cj}^2 \leq 1$, we have $|B| < \delta m$. Define non-normalized states

$$\begin{aligned}
|A(x)\rangle &= \sum_{(c,j)\in A} (-1)^{c\cdot C(x)_j}\alpha_{cj}|c\rangle|j\rangle \\
|B\rangle &= \sum_{(c,j)\in B} \alpha_{cj}|c\rangle|j\rangle.
\end{aligned}$$

The pure states $|A(x)\rangle + |B\rangle$ and $|A(x)\rangle - |B\rangle$ each correspond to a $y \in \{0,1\}^m$ that is corrupted (compared to $C(x)$) in at most $|B| \leq \delta m$ positions, so the decoder can recover x_i from each of these states. If x has $x_i = 1$, then we have:

$$\begin{aligned}
p(A(x)+B) &\geq 1/2+\varepsilon \\
p(A(x)-B) &\geq 1/2+\varepsilon.
\end{aligned}$$

Since $p(A\pm B) = p(A)+p(B)\pm(\langle A|D|B\rangle + \langle B|D|A\rangle)$, averaging the previous two inequalities gives

$$p(A(x)) + p(B) \geq 1/2+\varepsilon.$$

Similarly, if x' has $x'_i = 0$, then

$$p(A(x')) + p(B) \leq 1/2-\varepsilon.$$

Hence, for the normalized states $\frac{1}{a}|A(x)\rangle$ and $\frac{1}{a}|A(x')\rangle$:

$$p\left(\frac{1}{a}A(x)\right) - p\left(\frac{1}{a}A(x')\right) \geq 2\varepsilon/a^2.$$

Since this holds for every x, x' with $x_i = 1$ and $x'_i = 0$, there are constants $q_1, q_0 \in [0,1]$, $q_1 - q_0 \geq 2\varepsilon/a^2$, such that $p(\frac{1}{a}A(x)) \geq q_1$ whenever $x_i = 1$ and $p(\frac{1}{a}A(x)) \leq q_0$ whenever $x_i = 0$.

If we had a copy of the state $\frac{1}{a}|A(x)\rangle$, then we could run the procedure below to recover x_i. Here we assume that $q_1 \geq 1/2+\varepsilon/a^2$ (if not, then we must have $q_0 \leq 1/2-\varepsilon/a^2$ and we can use the same argument with 0 and 1 reversed), and that $q_1+q_0 \geq 1$ (if not, then $q_0 \leq 1/2-\varepsilon/a^2$ and we're already done).

Output 0 with probability $q = 1-1/(q_1+q_0)$,
and otherwise output the result of the decoder's 2-outcome measurement on $\frac{1}{a}|A(x)\rangle$.

If $x_i = 1$, then the probability that this procedure outputs 1 is

$$(1-q)p\left(\frac{1}{a}A(x)\right) \geq (1-q)q_1 = \frac{q_1}{q_1+q_0} = \frac{1}{2}+\frac{q_1-q_0}{2(q_1+q_0)} \geq \frac{1}{2}+\frac{\varepsilon}{2a^2}.$$

If $x_i = 0$, then the probability that the procedure outputs 0 is

$$q+(1-q)\left(1-p\left(\frac{1}{a}A(x)\right)\right) \geq q+(1-q)(1-q_0) = 1-\frac{q_0}{q_1+q_0} = \frac{q_1}{q_1+q_0} \geq \frac{1}{2}+\frac{\varepsilon}{2a^2}.$$

Thus we can recover x_i with good probability if we have the state $\frac{1}{a}|A(x)\rangle$ (which depends on i as well as x).

It remains to show how we can obtain $\frac{1}{a}|A(x)\rangle$ from $|U(x)\rangle$ with reasonable probability. This we do by applying a measurement with operators $M^\dagger M$ and $I-M^\dagger M$ to $|U(x)\rangle$, where $M = \sqrt{\delta m}\sum_{(c,j)\in A}\alpha_{cj}|c,j\rangle\langle c,j|$. Both $M^\dagger M$ and $I-M^\dagger M$ are positive operators (as required for a measurement) because $0 \leq \sqrt{\delta m}\alpha_{cj} \leq 1$ for all $(c,j) \in A$. The measurement gives the first outcome with probability

$$\langle U(x)|M^\dagger M|U(x)\rangle = \frac{\delta m}{2m}\sum_{cj\in A}\alpha_{cj}^2 = \frac{\delta a^2}{2}.$$

In this case we have obtained the normalized version of $M|U(x)\rangle$, which is $\frac{1}{a}|A(x)\rangle$. Suppose we have $r = 2/(\delta a^2)$ copies of $|U(x)\rangle$ and we do the measurement separately on each of them. Then with probability $1-(1-\delta a^2/2)^r \geq 1/2$, one of those will give the first outcome, in which case we can predict x_i with probability $\frac{1}{2}+\frac{\varepsilon}{2a^2}$. If all measurements give the second outcome then we just output a fair coin flip as our guess for x_i. Overall, our recovery probability is now

$$p \geq \frac{1}{2}\left(\frac{1}{2}+\frac{\varepsilon}{2a^2}\right)+\frac{1}{2}\cdot\frac{1}{2} = \frac{1}{2}+\frac{\varepsilon}{4a^2}.$$

Accordingly, r copies of the $(\log(m)+1)$-qubit state $|U(x)\rangle$ form a quantum random access code with recovery probability p. Using Theorem 2, $1-H(1/2+\eta) \geq 2\eta^2/\ln 2$, and $a^2 \leq 1$, gives

$$r(\log(m)+1) \geq (1-H(p))n \geq \frac{\varepsilon^2 n}{8a^4\ln 2} \geq \frac{\varepsilon^2 n}{8a^2\ln 2},$$

hence

$$\log m \geq \frac{\delta\varepsilon^2 n}{16\ln 2}-1.$$

□

3.3 Lower Bound for 2-Query LDCs

Theorem 4 *If* $C : \{0,1\}^n \to \{0,1\}^m$ *is a* $(2,\delta,\varepsilon)$*-locally decodable code, then*

$$m \geq 2^{cn-1},$$

for $c = 3\delta\varepsilon^2/(98\ln 2)$.

Proof. The theorem combines Theorem 1 and 3. Straightforwardly, this would give a constant of $\delta\varepsilon^2/(49\ln 2)$. We get the better constant claimed here by observing that the 1-query LQDC derived from the 2-query LDC actually has 1/3 of the overall squared amplitude on queries where the control bit c is zero (and all those α_{0j} are in A). Hence in the proof of Theorem 3, we can redefine "small amplitude" to $\alpha_{cj} \leq \sqrt{2/(3\delta m)}$, and still B will have at most δm elements because $\sum_{(c,j)\in B}\alpha_{cj}^2 \leq 2/3$. This in turns allows us to make M a factor $\sqrt{3/2}$ larger, which improves the probability of getting $\frac{1}{a}|A(x)\rangle$ from $|U(x)\rangle$ to $3\delta a^2/4$ and allows us to decrease r to $4/(3\delta a^2)$. This translates to a lower bound $\log m \geq 3\delta\varepsilon^2 n/(32\ln 2) - 1$. Combining that with Theorem 1 (which makes ε a factor 4/7 smaller) gives $c = 3\delta\varepsilon^2/(98\ln 2)$, as claimed. □

Remarks:

(1) Note that a $(2,\delta,\varepsilon)$-LDC with *adaptive* queries gives a $(2,\delta,\varepsilon/2)$-LDC with non-adaptive queries: if query q_1 would be followed by query q_2^0 or q_2^1 depending on the outcome of q_1, then we can just guess in advance whether to query q_1 and q_2^0, or q_1 and q_2^1. With probability 1/2, the second query will be the one we would have made in the adaptive case and we're fine, in the other case we just flip a coin, giving overall recovery probability $1/2(1/2+\varepsilon) + 1/2(1/2) = 1/2 + \varepsilon/2$. Thus we also get slightly weaker but still exponential lower bounds for *adaptive* 2-query LDCs.

(2) The constant $3/(98\ln 2)$ can be optimized a bit further by choosing the number r of copies a bit larger in the proof of Theorem 3 and by using Peter Høyer's 9/10-algorithm (Appendix A) instead of our 11/14-algorithm from Lemma 1. More interesting, however, is the question whether the quadratic dependence on ε can be improved.

(3) For a $(2,\delta,\varepsilon)$-LDC where the decoder's output is the XOR of its two queries, we can give a better reduction than in Theorem 1. Now the quantum decoder can query $\frac{1}{\sqrt{2}}(|1\rangle|1\rangle + |1\rangle|2\rangle)$, giving

$$\frac{1}{\sqrt{2}}\left((-1)^{a_1}|1\rangle|1\rangle + (-1)^{a_2}|1\rangle|2\rangle\right) = (-1)^{a_1}\frac{1}{\sqrt{2}}\left(|1\rangle|1\rangle + (-1)^{a_1\oplus a_2}|1\rangle|2\rangle\right),$$

and extract $a_1 \oplus a_2$ from this with certainty. Thus the recovery probability remains $1/2+\varepsilon$ instead of going down to $1/2+4\varepsilon/7$. Accordingly, we also get better lower bounds for 2-query LDCs where the output is the XOR of the two queries, with $c = \delta\varepsilon^2/(16\ln 2)$ in the exponent.

(4) The second part of our proof is a reduction from a Locally Quantum-Decodable Code to a "smooth" quantum code and then to a code where the distribution of the queries is uniform. This reduction is known for classical codes as well (see the next section). Hence, an alternative way to get the exponential lower bound on m would be first to invoke the result by Katz and Trevisan that reduces an LDC to a code with a uniform query distribution. We can reduce further to the case where the decoder outputs the XOR of the q queried bits. Starting with such a uniformly smooth code, we can then use our reduction from 2 classical queries to 1 quantum query without any loss in recovery probability (see Remark 3). After this reduction we immediately end up with a quantum random access code of $\log m$ qubits and we are done. However, this proof would give a worse dependence on δ and ε than our current result.

4 Extensions

In this section we give various extensions and variations of the lower bound of the previous section.

4.1 Non-Binary Alphabets

Here we extend our lower bounds for binary 2-query LDCs to the case of 2-query LDCs over larger alphabets. For simplicity we assume the alphabet is $\Sigma = \{0,1\}^\ell$, so a query to position j now returns an ℓ-bit string $C(x)_j$. The definition of (q, δ, ε)-LDC from Section 2.2 carries over immediately, with $d(C(x), y)$ now measuring the Hamming distance between $C(x) \in \Sigma^m$ and $y \in \Sigma^m$.

We will need the notion of *smooth* codes and their connection to LDCs as stated in [21].

Definition 3 $C : \{0,1\}^n \to \Sigma^m$ *is a* (q, c, ε)-smooth code *if there is a classical randomized decoding algorithm* A *such that*

1. *A makes at most q queries, non-adaptively.*
2. *For all x and i we have* $\Pr[A^{C(x)}(i) = x_i] \geq 1/2 + \varepsilon$.
3. *For all x, i, and j, the probability that on input i machine A queries index j is at most c/m.*

Note that smooth codes only require good decoding on codewords $C(x)$, not on y that are close to $C(x)$. Katz and Trevisan [21, Theorem 1] established the following connection:

Theorem 5 (Katz & Trevisan) *A (q, δ, ε)-LDC $C : \{0,1\}^n \to \Sigma^m$ is a $(q, q/\delta, \varepsilon)$-smooth code.*

A converse to Theorem 5 also holds: a (q, c, ε)-smooth code is a $(q, \delta, \varepsilon - c\delta)$-LDC, because the probability that the decoder queries one of δm corrupted positions is at most $(c/m)(\delta m) = c\delta$. Hence LDCs and smooth codes are essentially equivalent, for appropriate choices of the parameters.

To prove the exponential lower bound for LDCs over non-binary alphabet Σ, we will reduce a smooth code over Σ to a somewhat longer *binary* smooth code that works well *averaged over x*. Then, we will show a lower bound on such average-case binary smooth codes in a way very similar to the proof of Theorem 4. The following key lemma was suggested to us by Luca Trevisan [35].

Lemma 2 (Trevisan) *Let $C : \{0,1\}^n \to \Sigma^m$ be a $(2, c, \varepsilon)$-smooth code. Then there exists a $(2, c \cdot 2^\ell, \varepsilon/2^{2\ell})$-smooth code $C' : \{0,1\}^n \to \{0,1\}^{m \cdot 2^\ell}$ that is good* on average, *i.e., there is a decoder A such that for all $i \in [n]$*

$$\frac{1}{2^n} \sum_{x \in \{0,1\}^n} \Pr[A^{C'(x)}(i) = x_i] \geq \frac{1}{2} + \frac{\varepsilon}{2^{2\ell}}.$$

Proof. We form the new binary code C' by replacing each symbol $C(x)_j \in \Sigma$ of the old code by its Hadamard code, which consists of 2^ℓ bits. The length of $C'(x)$ is $m \cdot 2^\ell$ bits. The new decoding algorithm uses the same randomness as the old one. Let us fix the two queries $j, k \in [m]$ and the output function $f : \Sigma^2 \to \{0,1\}$ of the old decoder. We will describe a new decoding algorithm that is good for an average x and looks only at one bit of the Hadamard codes of each of $a = C(x)_j$ and $b = C(x)_k$.

First, if for this specific j, k, f we have $\Pr_x[f(a,b) = x_i] \leq 1/2$, then the new decoder just outputs a random bit, so in this case it is at least as good as the old one for an average x. Now consider the case $\Pr_x[f(a,b) = x_i] = 1/2 + \eta$ for some $\eta > 0$. Switching from the $\{0,1\}$-notation to the

$\{-1,1\}$-notation enables us to say that $E_x[f(a,b)\cdot x_i]=2\eta$. Viewing a and b as two ℓ-bit strings, we can represent f by its Fourier representation (see e.g. [6]): $f(a,b)=\sum_{S,T\subseteq[\ell]}\hat{f}_{S,T}\prod_{s\in S}a_s\prod_{t\in T}b_t$ and hence

$$\sum_{S,T}\hat{f}_{S,T}E_x\left[\prod_{s\in S}a_s\prod_{t\in T}b_t\cdot x_i\right]=E_x\left[\left(\sum_{S,T}\hat{f}_{S,T}\prod_{s\in S}a_s\prod_{t\in T}b_t\right)\cdot x_i\right]=E_x[f(a,b)\cdot x_i]=2\eta.$$

Averaging and using that $|\hat{f}_{S_0,T_0}|\leq 1$, it follows that there exist subsets S_0,T_0 such that

$$\left|E_x\left[\prod_{s\in S_0}a_s\prod_{t\in T_0}b_t\cdot x_i\right]\right|\geq \hat{f}_{S_0,T_0}E_x\left[\prod_{s\in S_0}a_s\prod_{t\in T_0}b_t\cdot x_i\right]\geq\frac{2\eta}{2^{2\ell}}.$$

Returning to the $\{0,1\}$-notation, we must have either

$$\Pr_x[(S_0\cdot a\oplus T_0\cdot b)=x_i]\geq 1/2+\eta/2^{2\ell}$$

or

$$\Pr_x[(S_0\cdot a\oplus T_0\cdot b)=x_i]\leq 1/2-\eta/2^{2\ell},$$

where $S_0\cdot a$ and $T_0\cdot b$ denote inner products mod 2 of ℓ-bit strings. Accordingly, either the XOR of the two bits $S_0\cdot a$ and $T_0\cdot b$, or its negation, predicts x_i with average probability $\geq 1/2+\eta/2^{2\ell}$. Both of these bits are in the binary code $C'(x)$. The c-smoothness of C translates into $c\cdot 2^{\ell}$-smoothness of C'. Averaging over the classical randomness (i.e. the choice of j,k, and f) gives the lemma. □

This lemma enables us to modify our proof of Theorem 4 so that it works for non-binary alphabets Σ:

Theorem 6 *If $C:\{0,1\}^n\to\Sigma^m=(\{0,1\}^\ell)^m$ is a $(2,\delta,\varepsilon)$-locally decodable code, then*

$$m\geq 2^{cn-\ell},$$

for $c=\Theta(\delta\varepsilon^2/2^{5\ell})$.

Proof. Using Theorem 5 and Lemma 2, we turn C into a binary $(2,2^{\ell+1}/\delta,\varepsilon/2^{2\ell})$-smooth code C' that has average recovery probability $1/2+\varepsilon/2^{2\ell}$ and length $m'=m\cdot 2^\ell$ bits. Since its decoder XORs its two binary queries, we can reduce this to one quantum query without any loss in the average recovery probability (see the third remark following Theorem 4).

We now reduce this quantum smooth code to a quantum random access code, by a modified version of the proof of Theorem 4. The smoothness of C' implies that all amplitudes α_j (which depend on i) in the one quantum query satisfy $\alpha_j\leq\sqrt{2^{\ell+1}/\delta m'}$. Hence there is no need to split the set of j's into A and B. Also, the control bit c will always be 1, so we can ignore it.

Consider the states $|U(x)\rangle=\frac{1}{\sqrt{m'}}\sum_{j=1}^{m'}(-1)^{C(x)'_j}|j\rangle$ and $|A(x)\rangle=\sum_{j=1}^{m'}\alpha_j(-1)^{C(x)'_j}|j\rangle$, and the 2-outcome measurement with operators $M=\sqrt{\delta m'/2^{\ell+1}}\sum_j\alpha_j|j\rangle\langle j|$ and $I-M$. The probability that the measurement takes us from $|U(x)\rangle$ to the renormalized $M|U(x)\rangle$ $(=|A(x)\rangle)$ is equal to $\langle U(x)|M^*M|U(x)\rangle=\delta/2^{\ell+1}$. Hence $r=2^{\ell+1}/\delta$ copies of $|U(x)\rangle$ forms a quantum random access code with average success probability

$$p\geq\frac{1}{2}\left(\frac{1}{2}+\frac{\varepsilon}{2^{2\ell}}\right)+\frac{1}{2}\cdot\frac{1}{2}=\frac{1}{2}+\frac{\varepsilon}{2^{2\ell+1}}.$$

The $(1 - H(p))n$ lower bound for a quantum random access code holds even if the recovery probability p is only an average over x, which gives

$$r \cdot \log(m') \geq (1 - H(p))n,$$

which implies the statement of the theorem. □

4.2 Bounds for More Than 2 Queries

Here we address the case of LDCs over the binary alphabet where the decoder asks more than 2 queries. There is no obvious way to extend our 2-to-1 reduction to more than 2 classical queries, since a quantum computer needs $\lceil q/2 \rceil$ queries to compute the parity of q bits with any advantage [4, 15]. In particular, it needs 2 quantum queries to compute the parity of 3 bits, and we don't have any lower bounds for 2-query LQDCs. Still, for LDCs with $q \geq 3$ queries we were able to improve the polynomial lower bounds $m = \Omega(n^{1+1/(q-1)})$ of Katz and Trevisan [21] somewhat:

Theorem 7 *If $C : \{0,1\}^n \to \{0,1\}^m$ is a (q, δ, ε)-locally decodable code, then*

$$m = \Omega\left(\left(\frac{n}{\log n}\right)^{1+1/(\lceil q/2 \rceil - 1)}\right),$$

where the constant under the $\Omega(\cdot)$ depends on q, δ and ε.

Proof. Suppose for simplicity that q is even and m is a multiple of q. By Theorem 5, it suffices to prove a bound for a (q, c, ε)-smooth code, with $c = q/\delta$. We will use the following result to make the smooth code uniform.

Fact (Katz & Trevisan [21, discussion in Section 4]): A (q, c, ε)-smooth code is a $(q, q, \varepsilon^2/2c)$-smooth code that is good on average. For every i, the new q-query decoder has a fixed partition M_i of $[m]$ into m/q q-tuples; it just picks a random q-tuple $(j_1, \ldots, j_q) \in M_i$ and outputs a Boolean function of the q bits $C(x)_{j_1}, \ldots, C(x)_{j_q}$. For every i, the decoding of x_i will be correct with probability at least $1/2 + \varepsilon^2/2c$ averaged over all x.

By a proof analogous to Lemma 2, we can ensure that the decoder actually computes the XOR of the q queried bits (or its negation). The average correctness probability will still be at least $1/2 + \varepsilon^2/c2^{q+1}$. We will derive a quantum random access code from this uniform smooth code. Let $P_{ij} = |i\rangle\langle i| + |j\rangle\langle j|$ be the projector on the states $|i\rangle$ and $|j\rangle$. Suppose $(i_1, j_1), \ldots, (i_{m/2}, j_{m/2})$ is a partition of all the q-tuples in M_i into pairs. By measuring the uniform state

$$|U(x)\rangle = \frac{1}{\sqrt{m}} \sum_{j=1}^{m} (-1)^{C(x)_j} |j\rangle$$

with operators $P_{i_1 j_1}, \ldots, P_{i_{m/2} j_{m/2}}$, we get

$$\frac{1}{\sqrt{2}} \left((-1)^{C(x)_{i_\ell}} |i_\ell\rangle + (-1)^{C(x)_{j_\ell}} |j_\ell\rangle \right),$$

for random $1 \leq \ell \leq m/2$. From this we can obtain $C(x)_{i_\ell} \oplus C(x)_{j_\ell}$, so we can generate the XOR of a random pair from the partition. In order to recover x_i we need to find $q/2$ different pairs that come from the same q-tuple.

Each state $|U(x)\rangle$ gives us a random pair out of the possible $m/2$. By the Birthday Paradox, if we have $O(m^{1-2/q})$ copies of the $\log m$-qubit state $|U(x)\rangle$, then with high probability we will find $q/2$ different pairs that come from the same q-tuple and hence be able to recover x_i. In other words, $O(m^{1-2/q})$ copies of the $\log m$-qubit state $|U(x)\rangle$ constitute an (average) random access code. The random access code lower bound (Appendix B) now gives

$$m^{1-2/q} \cdot \log m = \Omega(n),$$

which implies $m = \Omega((n/\log n)^{1+2/(q-2)})$. $\square$

For example, for $q = 4$ queries our lower bound is $m = \Omega((n/\log n)^2)$ while Katz and Trevisan have $m = \Omega(n^{4/3})$.

4.3 Locally Quantum-Decodable Codes with Few Queries

The third remark of Section 3.3 immediately generalizes to:

Theorem 8 *A $(2q, \delta, \varepsilon)$-LDC where the decoder's output is the XOR of the $2q$ queried bits, is a (q, δ, ε)-LQDC.*

LDCs with q queries can be obtained from q-server PIR schemes with 1-bit answers by concatenating the answers that the servers give to all possible queries of the user. Beimel et al. [8, Corollary 4.3] recently improved the best known upper bounds on q-query LDCs, based on their improved PIR construction. They give a general upper bound $m = 2^{n^{O(\log\log q/q \log q)}}$ for q-query LDCs, for some constant depending on δ and ε, as well as more precise estimates for small q. In particular, for $q = 4$ they construct an LDC of length $m = 2^{O(n^{3/10})}$. All their LDCs are of the XOR-type, so we can reduce the number of queries by half when allowing quantum decoding. For instance, their 4-query LDC is a 2-query LQDC with length $m = 2^{O(n^{3/10})}$. In contrast, any 2-query LDC needs length $m = 2^{\Omega(n)}$ as proved above.

For general LDCs we can do something nearly as good, using van Dam's result that a q-bit oracle can be recovered with probability nearly 1 using $q/2 + O(\sqrt{q})$ quantum queries [13]:

Theorem 9 *A (q, δ, ε)-LDC is a $(q/2 + O(\sqrt{q}), \delta, \varepsilon/2)$-LQDC.*

4.4 Locally Decodable Erasure Codes

Recently, the notion of a Locally Decodable Erasure Code (LDEC) was used in the construction of extractors [25, Section 3.1]. This is a code where, even if $(1-\varepsilon)m$ of all positions of the codeword are erased, we can still recover each x_i using only q queries to the remaining positions.

Definition 4 *Consider a map $C : \{0,1\}^n \to \Sigma^m$. We say that message position i is* decodable *from codeword positions $j_1, \ldots, j_q$ if there exists a function f such that $f(C(x)_{j_1}, \ldots, C(x)_{j_q}) = x_i$ for all x. C is a (q, ε)*-LDEC*, if for every i, in every ε-fraction of the positions of the codeword, there exists a q-tuple of positions from which i is decodable.*

Here we show that LDECs are equivalent to smooth codes, as defined in Section 4.1, and hence to LDCs. Consider some LDEC with codewords of length m. This equivalence shows that our lower bounds also hold for LDECs. In particular, $(2, \varepsilon)$-LDECs need exponential length.

First consider some LDEC. Take S to be the set of an ε-fraction of positions of the codeword. By definition, there exists a "good" q-tuple in S, i.e., one from which we can decode message position i. Remove these q positions of the codeword from S and replace them by some other q positions. Now in this new set S' of positions there should still be a "good" q-tuple. Remove it and go on. You can repeat this substitution $(1-\varepsilon)m/q$ times, where m is the size of the code. Therefore, there are $\Omega(m)$ disjoint q-tuples that are "good" for x_i and so the code is a smooth code: the smooth decoder just picks one of these tuples at random and queries it positions.

The converse is also true. A smooth code contains $\Omega(m)$ disjoint q-tuples, say βm of them, that are "good" for x_i. Hence, in any subset of the positions of the codeword of size $(1-\beta)m+1$, there exists a "good" q-tuple and therefore the code is an LDEC with $\varepsilon \approx 1-\beta$.

5 Private Information Retrieval

As mentioned, there is a close connection between locally decodable codes and private information retrieval. In this section we use a variant of our 2-to-1 reduction to prove new *lower bounds* for PIR and new *upper bounds* for QPIR.

5.1 Lower Bounds for Binary 2-Server PIR

To get lower bounds for 2-server PIRs with 1-bit answers, we again give a 2-step proof: a reduction of 2 classical servers to 1 quantum server, combined with a lower bound for 1-server quantum PIR.

Theorem 10 *If there exists a classical 2-server PIR scheme with t-bit queries, 1-bit answers, and recovery probability $1/2+\varepsilon$, then there exists a quantum 1-server PIR scheme with $(t+2)$-qubit queries, $(t+2)$-qubit answers, and recovery probability $1/2+4\varepsilon/7$.*

Proof. The proof is analogous to the proof for locally decodable codes (Theorem 1). If we let the quantum user employ the same randomness as the classical one, the problem boils down to computing some $f(a_1,a_2)$, where a_1 is the first server's 1-bit answer to query q_1, and a_2 is the second server's 1-bit answer to query q_2. However, in addition we now have to hide i from the quantum server. This we do by making the quantum user set up the $(4+t)$-qubit state

$$\frac{1}{\sqrt{3}}\left(|0\rangle|0,0^t\rangle + |1\rangle|1,q_1\rangle + |2\rangle|2,q_2\rangle\right),$$

where '0^t' is a string of t 0s. The user sends everything but the first register to the server. The state of the server is now a uniform mixture of $|0,0^t\rangle$, $|1,q_1\rangle$, and $|2,q_2\rangle$. By the security of the classical protocol, $|1,q_1\rangle$ contains no information about i (averaged over the user's randomness), and the same holds for $|2,q_2\rangle$. Hence the server gets no information about i.

The quantum server then puts $(-1)^{a_j}$ in front of $|j,q_j\rangle$ ($j\in\{1,2\}$), leaves $|0,0^t\rangle$ alone, and sends everything back. Note that we need to supply the name of the classical server $j\in\{1,2\}$ to tell the server in superposition whether it should play the role of server 1 or 2. The user now has

$$\frac{1}{\sqrt{3}}\left(|0\rangle|0,0^t\rangle + (-1)^{a_1}|1\rangle|1,q_1\rangle + (-1)^{a_2}|2\rangle|2,q_2\rangle\right).$$

From this we can compute $f(a_1,a_2)$ with success probability exactly $11/14$, giving overall recovery probability $1/2+4\varepsilon/7$ as in Theorem 1. $\square$

Combining the above reduction with the quantum random access code lower bound, we obtain the first $\Omega(n)$ lower bound that holds for all 1-bit-answer 2-server PIRs, not just for linear ones.

Theorem 11 *A classical 2-server PIR scheme with t-bit queries, 1-bit answers, and recovery probability $1/2+\varepsilon$, has $t \geq (1-H(1/2+4\varepsilon/7))n-2$.*

Proof. We first reduce the 2 classical servers to 1 quantum server in the way of Theorem 10. Now consider the state of the quantum PIR scheme after the user sends his $(t+2)$-qubit message $|\phi_i\rangle$:

$$\sum_r \sqrt{\frac{p_r}{3}}|r\rangle \left(|0\rangle|0,0^t\rangle + |1\rangle|1,q_1(r,i)\rangle + |2\rangle|2,q_2(r,i)\rangle\right).$$

Here the p_r are the classical probabilities of the user (these depend on i) and $q_j(r,i)$ is the t-bit query that the user sends to server j in the classical 2-server scheme, if he wants x_i and has random string r. Letting $B=\{0^{t+1}\}\cup\{1,2\}\times\{0,1\}^t$ be the server's basis states, we can write $|\phi_i\rangle$ as:

$$|\phi_i\rangle = \sum_{b\in B}\lambda_b|a_{ib}\rangle|b\rangle.$$

Here the $|a_{ib}\rangle$ are pure states that do not depend on x. The coefficients λ_b are non-negative reals that do not depend on i, for otherwise a measurement of b would give the server information about i, contradicting privacy. The server then tags on the appropriate phase s_{bx}, which is 1 for $b=0^{t+1}$ and $(-1)^{S_j(x,q_j)}$ for $b=jq_j$, $j\in\{1,2\}$. This gives

$$|\phi_{ix}\rangle = \sum_{b\in B}\lambda_b|a_{ib}\rangle s_{bx}|b\rangle.$$

Now the following pure state will be a random access code for x

$$|\psi_x\rangle = \sum_{b\in B}\lambda_b s_{bx}|b\rangle,$$

because a user can unitarily map $|0\rangle|b\rangle\mapsto|a_{ib}\rangle|b\rangle$ to map $|0\rangle|\psi_x\rangle\mapsto|\phi_{ix}\rangle$, from which he can get x_i with probability $p=1/2+4\varepsilon/7$ by completing the quantum PIR protocol. The state $|\psi_x\rangle$ has $t+2$ qubits, hence from Theorem 2 we obtain $t\geq(1-H(p))n-2$. □

For the special case where the classical PIR outputs the XOR of the two answer bits, we can improve our lower bound to $t\geq(1-H(1/2+\varepsilon))n-1$. In particular, $t\geq n-1$ in case of *perfect* recovery ($\varepsilon=1/2$), which is tight.

5.2 Lower Bounds for 2-Server PIR with Larger Answers

We can also extend our linear lower bound on 2-server PIR schemes with answer length $a=1$ (Theorem 11) to the case of 2-server PIR larger answer length. We use the translation from PIR to smooth codes given by Lemma 7.1 of Goldreich et al. [20]:

Lemma 3 (GKST) *If there is a classical 2-server PIR scheme with query length t, answer length a, and recovery probability $1/2+\varepsilon$, then there is a $(2,3,\varepsilon)$-smooth code $C:\{0,1\}^n\to\Sigma^m$ for $\Sigma=\{0,1\}^a$ and $m\leq 6\cdot 2^t$.*

Going through roughly the same steps as for the proof of Theorem 6, we obtain:

Theorem 12 *A classical 2-server PIR scheme with t-bit queries, a-bit answers, and recovery probability $1/2+\varepsilon$, has $t\geq\Omega(n\varepsilon^2/2^{5a})$.*

5.3 Lower Bounds for General 2-Server PIR

The previous lower bounds on the query length of 2-server PIR schemes were significant only for protocols with short answer length. Here we slightly improve the best known bound of $4 \log n$ [26] on the overall communication complexity of 2-server PIR schemes, by combining our Theorem 12 and Theorem 6 of Katz and Trevisan [21]. We restate their theorem here for the PIR setting. For the remainder of this section, we assume ε to be some fixed positive constant.

Theorem 13 (Katz & Trevisan) *Every 2-server PIR scheme with t-bit queries and a-bit answers has*

$$t \geq 2 \log \frac{n}{a} - O(1).$$

We now prove the following lower bound on the total communication $C = 2(t + a)$ of any 2-server PIR scheme with t-bit queries and a-bit answers:

Theorem 14 *Every 2-server PIR scheme has total communication*

$$C \geq (4.4 - o(1)) \log n.$$

Proof. We distinguish three cases, depending on the answer length of the scheme. Let $\delta = \log\log n / \log n$.

case 1: $a \leq (0.2 - \delta) \log n$. Then from Theorem 12 we get that $C \geq t = \Omega(n^{5\delta}) = \Omega((\log n)^5)$.

case 2: $(0.2 - \delta) \log n < a < 2.2 \log n$. Then from Theorem 13 we have

$$C = 2(t + a) > 2\left(2\log(n/(2.2\log n)) - O(1) + (0.2 - \delta)\log n\right) = (4.4 - o(1))\log n.$$

case 3: $a \geq 2.2 \log n$. Then obviously $C = 2(t + a) \geq 4.4 \log n$.

□

5.4 Upper Bounds for Quantum PIR

The best known LDCs are derived from classical PIR schemes with 1-bit answers where the output is the XOR of the 1-bit answers that the user receives. By allowing quantum queries, we can reduce the number of queries by half to obtain more efficient LQDCs. Similarly, we can also turn the underlying classical k-server PIRs directly into quantum PIRs with $k/2$ servers.

Most interestingly, there exists a 4-server PIR with 1-bit answers and communication complexity $O(n^{3/10})$ [8, Example 4.2]. This gives us a quantum 2-server PIR scheme with $O(n^{3/10})$ communication, improving upon the communication required by the best known classical 2-server PIR scheme, which has been $O(n^{1/3})$ ever since the introduction of PIR by Chor et al. [11]. In the introduction we mentioned also some quantum upper bounds for $k > 2$ servers, which are obtained similarly.

6 Conclusion and Open problems

This paper is the first where a new classical result is proved using techniques from quantum computing in an apparently essential way (at least, we don't know a classical proof of the same result). Clearly, it would be very interesting to find other such applications. This would much broaden the relevance of quantum computing and make it less conditional on whether an actual quantum computer will ever be built.

There are also many interesting open questions related to the tradeoffs between the various parameters in LDCs. In particular, it is still open whether one can achieve $m = poly(n)$ LDCs using only a constant (or even sublogarithmic) number of queries. We would like to obtain better lower bounds for $q > 2$ queries and explore the connections of LDCs to other combinatorial constructions.

Similarly, the main complexity questions about general PIR schemes are still wide open, even for the 2-server case if we don't restrict the answer size. The $O(n^{1/3})$-protocol of [11] has been the best known for a long time for the 2-server case, and it would be very nice to show that this is close to optimal. Finally, we exhibited 2-server quantum PIR schemes that are more efficient than the best known classical ones. It would be very interesting to improve these further, and to prove that QPIR is more efficient than the best (rather than the best known) classical PIR schemes.

Acknowledgments

We would like to thank Luca Trevisan for many insightful comments throughout this work and also for allowing us to include Lemma 2 in Section 4.1. We also thank Harry Buhrman, Richard Gill, Peter Høyer, Hartmut Klauck, Ashwin Nayak, Kenji Obata, Pranab Sen (and via him also Rahul Jain), Mario Szegedy, Ashish Thapliyal, John Tromp, and Stephanie Wehner for helpful discussions. We thank Amos Beimel for sending us a version of [8], Bill Gasarch for sending us a version of [7], Claude Crépeau and Nicolas Gisin for references to [19, 31], and the anonymous JCSS referee for many comments that improved the presentation of the paper.

References

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

[2] L. Babai, L. Fortnow, L. Levin, and M. Szegedy. Checking computations in polylogarithmic time. In *Proceedings of 23rd ACM STOC*, pages 21–31, 1991.

[3] L. Babai, L. Fortnow, N. Nisan, and A. Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3(4):307–318, 1993.

[4] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001. Earlier version in FOCS'98. quant-ph/9802049.

[5] D. Beaver and J. Feigenbaum. Hiding instances in multioracle queries. In *Proceedings of 7th Annual Symposium on Theoretical Aspects of Computer Science (STACS'90)*, volume 415 of *Lecture Notes in Computer Science*, pages 37–48. Springer, 1990.

[6] R. Beigel. The polynomial method in circuit complexity. In *Proceedings of the 8th IEEE Structure in Complexity Theory Conference*, pages 82–95, 1993.

[7] R. Beigel, L. Fortnow, and W. Gasarch. Nearly tight bounds for private information retrieval systems. Technical Report 2002-L001N, NEC Laboratories America, October 2002.

[8] A. Beimel, Y. Ishai, E. Kushilevitz, and J. Raymond. Breaking the $O(n^{1/(2k-1)})$ barrier for information-theoretic Private Information Retrieval. In *Proceedings of 43rd IEEE FOCS*, pages 261–270, 2002.

[9] E. Ben-Sasson, O. Goldreich, P. Harsha, M. Sudan, and S. Vadhan. Robust PCPs of proximity, shorter PCPs and applications to coding. To appear in *Proceedings of 36th ACM STOC*, 2004.

[10] H. Buhrman, P. B. Miltersen, J. Radhakrishnan, and S. Venkatesh. Are bitvectors optimal? In *Proceedings of 32nd ACM STOC*, pages 449–458, 2000.

[11] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. *Journal of the ACM*, 45(6):965–981, 1998. Earlier version in FOCS'95.

[12] R. Cleve, W. van Dam, M. Nielsen, and A. Tapp. Quantum entanglement and the communication complexity of the inner product function. In *Proceedings of 1st NASA QCQC conference*, volume 1509 of *Lecture Notes in Computer Science*, pages 61–74. Springer, 1998. quant-ph/9708019.

[13] W. van Dam. Quantum oracle interrogation: Getting all information for almost half the price. In *Proceedings of 39th IEEE FOCS*, pages 362–367, 1998. quant-ph/9805006.

[14] A. Deshpande, R. Jain, T. Kavitha, S. Lokam, and J. Radhakrishnan. Better lower bounds for locally decodable codes. In *Proceedings of 17th IEEE Conference on Computational Complexity*, pages 184–193, 2002.

[15] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. A limit on the speed of quantum computation in determining parity. *Physical Review Letters*, 81:5442–5444, 1998. quant-ph/9802045.

[16] J. Feigenbaum and L. Fortnow. Random-self-reducibility of complete sets. *SIAM Journal on Computing*, 22(5):994–1005, 1993. Earlier version in Structures'91.

[17] P. Gemmell, R. Lipton, R. Rubinfeld, M. Sudan, and A. Wigderson. Self-testing/correcting for polynomials and for approximate functions. In *Proceedings of 23rd ACM STOC*, pages 32–42, 1991.

[18] P. Gemmell and M. Sudan. Highly resilient correctors for polynomials. *Information Processing Letters*, 43(4):169–174, 1992.

[19] N. Gisin, R. Renner, and S. Wolf. Linking classical and quantum key agreement: Is there a classical analog to bound entanglement? *Algorithmica*, 34(4):389–412, 2002. Earlier version in Crypto'2000.

[20] O. Goldreich, H. Karloff, L. Schulman, and L. Trevisan. Lower bounds for linear locally decodable codes and private information retrieval. In *Proceedings of 17th IEEE Conference on Computational Complexity*, pages 175–183, 2002. Also on ECCC.

[21] J. Katz and L. Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *Proceedings of 32nd ACM STOC*, pages 80–86, 2000.

[22] I. Kerenidis and R. de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. In *Proceedings of 35th ACM STOC*, pages 106–115, 2003. quant-ph/0208062.

[23] H. Klauck, A. Nayak, A. Ta-Shma, and D. Zuckerman. Interaction in quantum communication and the complexity of set disjointness. In *Proceedings of 33rd ACM STOC*, pages 124–133, 2001.

[24] R. Lipton. New directions in testing. In *Vol. 2 of Series in Discrete Mathematics and Theoretical Computer Science*, pages 191–202. ACM/AMS, 1991.

[25] C. Lu, O. Reingold, S. Vadhan, and A. Wigderson. Extractors: Optimal up to constant factors. In *Proceedings of 35th ACM STOC*, pages 602–611, 2003.

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

[27] A. Nayak. Optimal lower bounds for quantum automata and random access codes. In *Proceedings of 40th IEEE FOCS*, pages 369–376, 1999. quant-ph/9904093.

[28] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[29] K. Obata. Optimal lower bounds for 2-query locally decodable linear codes. In *Proceedings of 6th RANDOM*, volume 2483 of *Lecture Notes in Computer Science*, pages 39–50, 2002.

[30] J. Radhakrishnan, P. Sen, and S. Venkatesh. The quantum complexity of set membership. In *Proceedings of 41st IEEE FOCS*, pages 554–562, 2000. quant-ph/0007021.

[31] R. Renner and S. Wolf. New bounds in secret-key agreement: The gap between formation and secrecy extraction. In *Proceedings of Eurocrypt'03*, volume 2656 of *Lecture Notes in Computer Science*, pages 562–577. Springer, 2003.

[32] P. Sen and S. Venkatesh. Lower bounds in the quantum cell probe model. In *Proceedings of 28th ICALP*, volume 2076 of *Lecture Notes in Computer Science*, pages 358–369. Springer, 2001. More extensive version at quant-ph/0104100.

[33] M. Sipser and D. A. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42:1710–1722, 1996. Earlier version in FOCS'94.

[34] M. Sudan, L. Trevisan, and S. Vadhan. Pseudorandom generators without the XOR lemma. In *Proceedings of 31st ACM STOC*, pages 537–546, 1999.

[35] L. Trevisan. Personal communication, September 2002.

A Optimal 1-Query Quantum Algorithms for 2-Bit Functions

In this appendix we show that every 2-bit Boolean function f can be computed with success probability 9/10 using only one quantum query, and that this is optimal for functions like AND

and OR.[3] If f is constant or depends on only one of its 2 input bits x_1 and x_2, then we can obviously compute it with one query. If f is PARITY or its negation, then it is well known that f can be computed exactly with one quantum query. The only remaining case is where f is an imbalanced function, i.e. has one 1-input and three 0-inputs, or vice versa. These 8 possible functions are all equivalent, so we will restrict attention to the NOR-function, which is 1 iff $x = 00$.

Peter Høyer discovered the following algorithm for doing the 2-bit NOR with 1 quantum query and error probability $\varepsilon = 1/10$. Using one quantum query we can obtain the state

$$\frac{1}{\sqrt{3}}\left(|0\rangle + (-1)^{x_1}|1\rangle + (-1)^{x_2}|2\rangle\right).$$

We now use a 2-outcome measurement where the first operator is the projection on the uniform superposition. We output 1 iff the measurement gives the first outcome. This has error probability 0 on the $x = 00$ input (where NOR $= 1$), and error probability 1/9 on each of the three other inputs. We can balance this to an algorithm with 2-sided error 1/10, by producing output 0 with probability 1/10, and running the above 1-query algorithm with probability 9/10.

We will now prove that his error $\varepsilon = 1/10$ is optimal. By the analysis of [4], the amplitudes of the final state of a 1-query quantum algorithm are degree-1 polynomials in the input variables, so the acceptance probability of the algorithm is a polynomial

$$p(x_1, x_2) = \sum_j |a_j + b_j(-1)^{x_1} + c_j(-1)^{x_2}|^2,$$

where j ranges over all basis states that would yield a 1 as output, and the a_j, b_j, c_j are complex numbers that are independent of the input. Let $a = (a_j)$ be the vector of a_js, $\| a \| = \sqrt{\langle a|a\rangle}$ its Euclidean norm, and similarly for b and c. If the algorithm has error probability $\leq \varepsilon$, then we have the following four conditions, one for each of the possible inputs:

$$\begin{array}{lcclcll}
\text{(A)} & 1-\varepsilon & \leq & p(0,0) & = & \| a+b+c \|^2 & \\
\text{(B)} & & & p(0,1) & = & \| a+b-c \|^2 & \leq \ \varepsilon \\
\text{(C)} & & & p(1,0) & = & \| a-b+c \|^2 & \leq \ \varepsilon \\
\text{(D)} & & & p(1,1) & = & \| a-b-c \|^2 & \leq \ \varepsilon
\end{array}$$

Averaging (B) and (C) gives

$$\| a \|^2 + \| b-c \|^2 \leq \varepsilon, \text{ hence } \| a \| \leq \sqrt{\varepsilon}.$$

Triangle inequality and (D) gives

$$\| b+c \| - \| a \| \leq \| a-b-c \| \leq \sqrt{\varepsilon}, \text{ hence } \| b+c \| \leq \| a \| + \sqrt{\varepsilon} \leq 2\sqrt{\varepsilon}.$$

Subtracting (D) from (A), and using Cauchy-Schwarz, gives

$$1 - 2\varepsilon \leq 4|\langle a|b+c\rangle| \leq 4\| a \| \cdot \| b+c \| \leq 4 \cdot \sqrt{\varepsilon} \cdot 2\sqrt{\varepsilon} = 8\varepsilon,$$

hence $\varepsilon \geq 1/10$.

[3]Unlike our 11/14 solution in Lemma 1, the query of the optimal 9/10 algorithm will depend on f. This means that we cannot directly use this algorithm in the PIR-context, as the query could leak information about f (and hence possibly about i) to the server.

B Lower Bound for Quantum Random Access Codes

As defined in Section 3.2, a quantum random access code is an encoding $x \mapsto \rho_x$, such that any bit x_i can be recovered with some probability $p \geq 1/2 + \varepsilon$ from ρ_x. Below we reprove Nayak's [27] linear lower bound on the length m of such encodings.

We assume familiarity with the following notions from quantum information theory, see [28, Chapters 11 and 12] for details. Very briefly, if we have a bipartite quantum system AB (given by some density matrix), then we use A and B to denote the states (reduced density matrices) of the individual systems. $S(A) = -\mathrm{Tr}(A \log A)$ is the *(Von Neumann) entropy* of A, which is the Shannon entropy of the probability distribution given by the eigenvalues of A. $S(A|B) = S(AB) - S(B)$ is the *conditional entropy* of A given B; and $S(A:B) = S(A) + S(B) - S(AB) = S(A) - S(A|B)$ is the *mutual information* between A and B.

We define an $n+m$-qubit state XM as follows:

$$\frac{1}{2^n} \sum_{x \in \{0,1\}^n} |x\rangle\langle x| \otimes \rho_x.$$

We use X to denote the first subsystem, X_i for its individual bits, and M for the second subsystem. By [28, Theorem 11.8.4] we have

$$S(XM) = n + \frac{1}{2^n} \sum_x S(\rho_x) \geq n = S(X).$$

Since M has m qubits we have $S(M) \leq m$, hence

$$S(X:M) = S(X) + S(M) - S(XM) \leq S(M) \leq m.$$

Using a chain rule for relative entropy, and the (highly non-trivial) subadditivity of Von Neumann entropy we get

$$S(X|M) = \sum_{i=1}^{n} S(X_i|X_1 \dots X_{i-1}M) \leq \sum_{i=1}^{n} S(X_i|M).$$

Since we can predict X_i from M with success probability p, Fano's inequality implies

$$H(p) \geq S(X_i|M).$$

In fact, Fano's inequality even applies under the weaker assumption that the success probability in predicting x_i is p only when *averaged* over all x. Putting the above equations together we obtain

$$(1 - H(p))n \leq S(X) - \sum_{i=1}^{n} S(X_i|M) \leq S(X) - S(X|M) = S(X:M) \leq m.$$