Corruption and Recovery-Efficient Locally Decodable Codes

David Woodruff

IBM Almaden
dpwoodru@us.ibm.com

Abstract. A (q, δ, ϵ)-*locally decodable code (LDC)* $C : \{0,1\}^n \rightarrow \{0,1\}^m$ is an encoding from n-bit strings to m-bit strings such that each bit x_k can be recovered with probability at least $\frac{1}{2} + \epsilon$ from $C(x)$ by a randomized algorithm that queries only q positions of $C(x)$, even if up to δm positions of $C(x)$ are corrupted. If C is a linear map, then the LDC is linear. We give improved constructions of LDCs in terms of the corruption parameter δ and recovery parameter ϵ. The key property of our LDCs is that they are *non-linear*, whereas all previous LDCs were linear.

1. For any $\delta, \epsilon \in [\Omega(n^{-1/2}),\ O(1)]$, we give a family of $(2, \delta, \epsilon)$-LDCs with length $m = \text{poly}(\delta^{-1}, \epsilon^{-1}) \exp(\max(\delta, \epsilon)\delta n)$. For linear $(2, \delta, \epsilon)$-LDCs, Obata has shown that $m \geq \exp(\delta n)$. Thus, for small enough constants δ, ϵ, two-query non-linear LDCs are shorter than two-query linear LDCs.
2. We improve the dependence on δ and ϵ of all constant-query LDCs by providing general transformations to non-linear LDCs. Taking Yekhanin's linear $(3, \delta, 1/2 - 6\delta)$-LDCs with $m = \exp\left(n^{1/t}\right)$ for any prime of the form $2^t - 1$, we obtain non-linear $(3, \delta, \epsilon)$-LDCs with $m = \text{poly}(\delta^{-1}, \epsilon^{-1}) \exp\left((\max(\delta, \epsilon)\delta n)^{1/t}\right)$.

Now consider a (q, δ, ϵ)-LDC C with a decoder that has n matchings $M_1, \ldots, M_n$ on the complete q-uniform hypergraph, whose vertices are identified with the positions of $C(x)$. On input $k \in [n]$ and received word y, the decoder chooses $e = \{a_1, \ldots, a_q\} \in M_k$ uniformly at random and outputs $\bigoplus_{j=1}^{q} y_{a_j}$. All known LDCs and ours have such a decoder, which we call a matching sum decoder. We show that if C is a two-query LDC with such a decoder, then $m \geq \exp(\max(\delta, \epsilon)\delta n)$. Interestingly, our techniques used here can further improve the dependence on δ of Yekhanin's three-query LDCs. Namely, if $\delta \geq 1/12$ then Yekhanin's three-query LDCs become trivial (have recovery probability less than half), whereas we obtain three-query LDCs of length $\exp\left(n^{1/t}\right)$ for any prime of the form $2^t - 1$ with non-trivial recovery probability for any $\delta < 1/6$.

1 Introduction

Classical error-correcting codes allow one to encode an n-bit message x into a codeword $C(x)$ such that even if a constant fraction of the bits in $C(x)$ are corrupted, x can still be recovered. It is well-known how to construct codes C of length $O(n)$ that can tolerate a constant fraction of errors, even in such a way that allows decoding in linear time [1]. However, if one is only interested in recovering a few bits of the message, then these codes have the disadvantage that they require reading all (or most) of the codeword. This motivates the following definition.

Definition 1. *([2]) Let $\delta, \epsilon \in [0,1]$, q an integer. We say $C : \{0,1\}^n \to \{0,1\}^m$ is a (q,δ,ϵ)-*locally decodable code *(LDC for short) if there is a probabilistic oracle machine A such that:*

- *In every invocation, A makes at most q queries.*
- *For every $x \in \{0,1\}^n$, every $y \in \{0,1\}^m$ with $\Delta(y, C(x)) \leq \delta m$, and every $k \in [n]$, $\Pr[A^y(k) = x_k] \geq \frac{1}{2} + \epsilon$, where the probability is taken over the internal coin tosses of A. An algorithm A satisfying the above is called a (q,δ,ϵ)-local decoding algorithm for C (a decoder for short).*

In the definition above, $\Delta(y, C(x))$ denote the Hamming distance between y and $C(x)$, that is, the number of coordinates for which the strings differ. For a (q,δ,ϵ)-LDC, we shall refer to q as *the number of queries*, δ as *the corruption parameter*, ϵ as *the recovery parameter*, and m as *the length.* An LDC is *linear* if C is a linear transformation over $GF(2)$. Note that recovery probability $1/2$ (corresponding to $\epsilon = 0$) is trivial since the decoder can just flip a random coin.

There is a large body of work on locally decodable codes. Katz and Trevisan [2] formally defined LDCs, proved that 1-query LDCs do not exist, and proved super-linear lower bounds on the length of constant-query LDCs. We refer the reader to the survey [3] and the references therein.

All known constructions of LDCs with a constant number of queries are super-polynomial in length, and not even known to be of subexponential length. Thus, understanding the asymptotics in the exponent of the length of such codes is important, and could be useful in practice for small values of n. A lot of work has been done to understand this exponent for two-query linear LDCs [4–7]. Important practical applications of LDCs include private information retrieval and load-balancing in the context of distributed storage. Depending on the parameters of the particular application, δ and ϵ may be flexible, and our constructions will be able to exploit this flexibility.

We state the known bounds relevant to this paper. The first two concern LDCs for which $q = 2$, while the remaining pertain to $q > 2$.

Notation: $\exp(f(n))$ denotes a function $g(n)$ that is $2^{O(f(n))}$.

Theorem 1. *([8])*[1] *Any $(2, \delta, \epsilon)$-LDC satisfies $m \geq \exp(\epsilon^2 \delta n)$.*

For linear LDCs, a tight lower bound is known.

Theorem 2. *([6, 7]) Any linear $(2, \delta, \epsilon)$-LDC has $m \geq \exp(\delta n/(1-2\epsilon))$. Moreover, there exists a linear $(2, \delta, \epsilon)$-LDC with $m \leq \exp(\delta n/(1-2\epsilon))$.*

The shortest LDCs for small values of $q > 2$ are due to Yekhanin [9], while for large values one can obtain the shortest LDCs by using the LDCs of Yekhanin together with a recursion technique of Beimel, Ishai, Kushilevitz, and Raymond [10]. The following is what is known for $q = 3$.

Theorem 3. *([9]) For any $\delta \leq 1/12$ and any prime of the form $2^t - 1$, there is a linear $(3, \delta, 1/2 - 6\delta)$-LDC with $m = \exp\left(n^{1/t}\right)$. Using the largest known such prime, this is $m = \exp\left(n^{1/32582657}\right)$.*

Notice that this theorem does not allow one to obtain shorter LDCs for small δ and $\epsilon < 1/2 - 6\delta$, as intuitively should be possible.

Results: We give improved constructions of constant-query LDCs in terms of the corruption parameter δ and recovery parameter ϵ. A key property of our LDCs is that they are the first non-linear LDCs. Our main theorem is the following transformation.

Theorem 4. *Given a family of $(q, \delta, 1/2 - \beta\delta)$-LDCs of length $m(n)$, where $\beta > 0$ is any constant, and $\delta < 1/(2\beta)$ is arbitrary (i.e., for a given n, the same encoding function C is a $(q, \delta, 1/2 - \beta\delta)$-LDC for any $\delta < 1/(2\beta)$), there is a family of non-linear $(q, \Theta(\delta), \epsilon)$-LDCs of length $O(dr^2)m(n'/r)$ for any $\delta, \epsilon \in [\Omega(n^{-1/2}),\ O(1)]$, where $d = \max(1, O(\epsilon/\delta))$, $r = O((\epsilon + \delta)^{-2})$, and $n' = n/d$.*

As a corollary, for any $\delta, \epsilon \in [\Omega(n^{-1/2},\ O(1)]$, we give a $(2, \delta, \epsilon)$-LDC with length $m = \text{poly}(\delta^{-1}, \epsilon^{-1}) \exp(\max(\delta, \epsilon)\delta n)$. Thus, by Theorem 2, as soon as δ and ϵ are small enough constants, this shows that 2-query non-linear LDCs are shorter than 2-query linear LDCs. This is the first progress on the question of Kerenidis and de Wolf [8] as to whether the dependence on δ and ϵ could be improved. Another corollary is that for any prime of the form $2^t - 1$ and any $\delta, \epsilon \in [\Omega(n^{-1/2}),\ O(1)]$, there is a family of non-linear $(3, \delta, \epsilon)$-LDCs with $m = \text{poly}(\delta^{-1}, \epsilon^{-1}) \exp\left((\max(\delta, \epsilon)\delta n)^{1/t}\right)$.

[1] This bound can be strengthened to $m \geq \exp\left(\epsilon^2 \delta n/(1-2\epsilon)\right)$ using the techniques of [6] in a relatively straightforward way. We do not explain the proof here, as our focus is when ϵ is bounded away from $1/2$, in which case the bounds are asymptotically the same.

Next, we show that our bound for 2-query LDCs is tight, up to a constant factor in the exponent, for a large family of LDCs including all known ones as well as ours. Let C be a (q,δ,ϵ)-LDC with a decoder that has n matchings $M_1,\ldots,M_n$ on the complete q-uniform hypergraph whose vertices are identified with the positions of $C(x)$. On input $k \in [n]$ and received word y, the decoder chooses a hyperedge $e = \{a_1,\ldots,a_q\} \in M_k$ uniformly at random and outputs $\bigoplus_{j=1}^{q} y_{a_j}$. We call such a decoder a *matching sum decoder*, and show that if a 2-query LDC C has such a decoder then $m \geq \exp(\max(\delta,\epsilon)\delta n)$. Thus, our upper bound is tight for such LDCs. To prove that for *any* $(2,\delta,\epsilon)$-LDC, $m \geq \exp(\max(\delta,\epsilon)\delta n)$, our result implies that it suffices to transform any LDC into one which has a matching sum decoder, while preserving δ, ϵ, and m up to small factors.

Finally, as an independent application of our techniques, we transform the $(3,\delta,1/2-6\delta)$-LDCs with $m = \exp(n^{1/t})$ of Theorem 3, into $(3,\delta,1/2-3\delta-\eta)$-LDCs with $m = \exp(n^{1/t})$, where $\eta > 0$ is an arbitrarily small constant. In particular, we extend the range of δ for which the LDCs in Theorem 3 become non-trivial from $\delta \leq 1/12$ to $\delta < 1/6$. Moreover, there is no 3-query LDC with a matching sum decoder with $\delta \geq 1/6$. Indeed, if the adversary corrupts exactly $m/6$ hyperedges of M_i, the recovery probability can be at most $1/2$.

Techniques: Our main idea for introducing non-linearity is the following. Suppose we take the message $x = x_1,\ldots,x_n$ and partition it into n/r blocks $B_1,\ldots,B_{n/r}$, each containing $r = \Theta(\epsilon^{-2})$ different x_i. We then compute $z_j = \text{majority}(x_i \mid i \in B_j)$, and encode the bits $z_1,\ldots,z_{n/r}$ using a (q,δ,ϵ)-LDC C. To obtain x_k if $x_k \in B_j$, we use the decoder for C to recover z_j with probability at least $1/2+\epsilon$. We should expect that knowing z_j is useful, since, using the properties of the majority function, $\Pr_x[x_k = z_j] \geq \frac{1}{2}+\epsilon$.

This suggests an approach: choose $s_1,\ldots,s_\tau \in \{0,1\}^n$ for a certain $\tau = O(r^2)$, apply the above procedure to each of $x \oplus s_1,\ldots,x \oplus s_\tau$, then take the concatenation. The $s_1,\ldots,s_\tau$ are chosen randomly so that for any $x \in \{0,1\}^n$ and any index k in any block B_j, a $\frac{1}{2}+\epsilon$ fraction of the different $x \oplus s_i$ have the property that their k-th coordinate agrees with the majority of the coordinates in B_j. The length of the encoding is now τm, where m is the length required to encode n/r bits.

To illustrate how recovery works, suppose that C were the Hadamard code. The decoder would choose a random $i \in [\tau]$ and decode the portion of the encoding corresponding to the (corrupted) encoding of $x \oplus s_i$. One could try to argue that with probability at least $1-2\delta$, the chosen positions by the Hadamard decoder are correct, and given that these are correct, $(x \oplus s_i)_k$ agrees with the majority of the coordinates in the associated block with probability at least $\frac{1}{2} + \epsilon$. If these events were independent, the success probability would be $\geq (1-2\delta)(1/2+\epsilon)+2\delta(1/2-\epsilon) = 1/2+\Omega(\epsilon)$.

However, these events are very far from being independent! Indeed, the adversary may first recover x from the encoding, and then for any given k, determine exactly which $(x \oplus s_i)_k$ agree with the majority of the coordinates in the associated block, and corrupt only these positions. This problem is unavoidable. However, we observe that we can instead consider $r = \Theta(\delta^{-2})$. Then, if $\delta = \Omega(\epsilon)$, we can show the decoder's success probability is at least $1/2 + \Omega(\epsilon)$. If, on the other hand, $\epsilon = \Omega(\delta)$, we can first allow δ to grow to $\Theta(\epsilon)$ via a technique similar to the upper bound given in [6], reducing n to $n' = \delta n/\epsilon$. Then we can effectively perform the above procedure with $r = \Theta(\epsilon^{-2})$ and $n'/r = \Theta(\epsilon^2 n') = \Theta(\epsilon\delta n)$.

To show that this technique is optimal for LDCs C with matching sum decoders, we need to significantly generalize the quantum arguments of [8]. A general matching sum decoder may have matchings M_i with very different sizes and contain edges that are correct for a very different number of $x \in \{0,1\}^n$. If we recklessly apply the techniques of [8], we cannot hope to obtain an optimal dependence on δ and ϵ.

Given such a C, we first apply a transformation to obtain a slightly longer LDC C' in which all matchings have the same size, and within a matching, the average fraction of x for which an edge is correct, averaged over edges, is the same for all matchings. We then apply another transformation to obtain an LDC C'' which increases the length of the code even further, but makes the matching sizes very large. Finally, we use quantum information theory to lower bound the length of C'', generalizing the arguments of [8] to handle the case when the average fraction of x for which an edge is correct, averaged over edges in a matching of C'', is sufficiently large.

Finally, we use an idea underlying the transformation from C' to C'' in our lower bound argument to transform the LDCs of Theorem 3 into LDCs with a better dependence on δ and ϵ, thereby obtaining a better upper bound. The idea is to blow up the LDC by a constant factor in the exponent, while increasing the sizes of the underlying matchings. Constructing the large matchings in the blown-up LDC is more complicated than it was in our lower bound argument, due to the fact that we run into issues of consistently grouping vertices of hypergraphs together which did not arise when we were working with graphs.

Other Related Work: Other examples where non-linear codes were shown to have superior parameters to linear codes include the construction of t-resilient functions [11, 12], where it is shown [13] that non-linear Kerdock codes outperform linear codes in the construction of such functions. See [14] for a study of non-linearity in the context of secret sharing.

2 Preliminaries

The following theorem is easy to prove using elementary Fourier analysis. We defer the proof to the full version. Throughout, we shall let c be the constant $(2/\pi)^{3/4}/4$.

Theorem 5. *Let r be an odd integer, and let $f : \{0,1\}^r \rightarrow \{0,1\}$ be the majority function, where $f(x) = 1$ iff there are more 1s than 0s in x. Then for any $k \in [r]$, $\Pr_{x \in \{0,1\}^r}[f(x) = x_k] > \frac{1}{2} + \frac{2c}{r^{1/2}}$.*

We also need an approximate version of this theorem, which follows from a simple application of the probabilistic method.

Lemma 1. *Let r and f be as in Theorem 5. Then there are $\tau = O(r^2)$ strings $\mu_1, \mu_2, \ldots, \mu_\tau \in \{0,1\}^r$ so that for all $x \in \{0,1\}^r$ and all $k \in [r]$, $\Pr_{i \in [\tau]}[f(x \oplus \mu_i) = (x \oplus \mu_i)_k] \geq \frac{1}{2} + \frac{c}{r^{1/2}}$.*

In our construction we will use the Hadamard code $C : \{0,1\}^n \rightarrow \{0,1\}^{2^n}$, defined as follows. Identify the 2^n positions of the codeword with distinct vectors $v \in \{0,1\}^n$, and set the vth position of $C(x)$ to $\langle v, x \rangle \bmod 2$. To obtain x_k from a vector y which differs from $C(x)$ in at most a δ fraction of positions, choose a random vector v, query positions y_v and $y_{v \oplus e_k}$, and output $y_v \oplus y_{v \oplus e_k}$. With probability at least $1 - 2\delta$, we have $y_v = \langle v, x \rangle$ and $y_{v \oplus e_k} = \langle v \oplus e_k, x \rangle$, and so $y_v \oplus y_{v \oplus e_k} = x_k$. It follows that for any $\delta > 0$, the Hadamard code is a $(2, \delta, 1/2 - 2\delta)$-LDC with $m = \exp(n)$.

Finally, in our lower bound, we will need some concepts from quantum information theory. We borrow notation from [8]. For more background on quantum information theory, see [15].

A *density matrix* is a positive semi-definite (PSD) complex-valued matrix with trace 1. A *quantum measurement* on a density matrix ρ is a collection of PSD matrices $\{P_j\}$ satisfying $\sum_j P_j^\dagger P_j = I$, where I is the identity matrix ($A^\dagger$ is the conjugate-transpose of A). The set $\{P_j\}$ defines a probability distribution X on indices j given by $\Pr[X = j] = \mathrm{tr}(P_j^\dagger P_j \rho)$.

We use the notation AB to denote a bipartite quantum system, given by some density matrix ρ^{AB}, and A and B to denote its subsystems. More formally, the density matrix of ρ^A is $\mathrm{tr}_B(\rho^{AB})$, where tr_B is a map known as the *partial trace* over system B. For given vectors $|a_1\rangle$ and

$|a_2\rangle$ in the vector space of A, and $|b_1\rangle$ and $|b_2\rangle$ in the vector space of B, $\mathrm{tr}_B(|a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2|) \overset{\mathrm{def}}{=} |a_1\rangle\langle a_2|\mathrm{tr}(|b_1\rangle\langle b_2|)$, and $\mathrm{tr}_B(\rho^{AB})$ is then well-defined by requiring tr_B to be a linear map.

$S(A)$ is the *von Neumann entropy* of A, defined as $\sum_{i=1}^{d} \lambda_i \log_2 \frac{1}{\lambda_i}$, where the λ_i are the eigenvalues of A. $S(A \mid B) = S(AB) - S(B)$ is the *conditional entropy* of A given B, and $S(A; B) = S(A) + S(B) - S(AB) = S(A) - S(A \mid B)$ is the *mutual information* between A and B.

3 The Construction

Let $C : \{0,1\}^n \to \{0,1\}^{m(n)}$ come from a family of $(q, \delta, 1/2 - \beta\delta)$-LDCs, where $\beta > 0$ is any constant, and $\delta < 1/(2\beta)$ is arbitrary (i.e., for a given n, the same function C is a $(q, \delta, 1/2 - \beta\delta)$-LDC for any $\delta < 1/(2\beta)$). For example, for any $\delta < 1/4$, the Hadamard code is a $(2, \delta, 1/2 - 2\delta)$-LDC, while Yekhanin [9] constructed a $(3, \delta, 1/2 - 6\delta)$-LDC for any $\delta < 1/12$.

Setup: Assume that $\delta, \epsilon \in [\Omega(n^{-1/2}),\ O(1)]$. W.l.o.g., assume n is a sufficiently large power of 3. Recall from Section 2 that we will use c to denote the constant $(2/\pi)^{3/4}/4$. Define the parameter $r = (\epsilon(1 + 2\beta c)/c + 2\beta\delta/c)^{-2} = \Theta((\epsilon + \delta)^{-2})$. Let $\tau = O(r^2)$ be as in Lemma 1. We define $d = \max(1, c\epsilon/\delta)$. Let $n' = n/d$. We defer the proof of the following lemma to the full version. The lemma establishes certain integrality and divisibility properties of the parameters that we are considering.

Lemma 2. *Under the assumption that $\delta, \epsilon \in [\Omega(n^{-1/2}),\ O(1)]$ and $\beta = \Theta(1)$, by multiplying δ and ϵ by positive constant factors, we may assume that the following two conditions hold simultaneously: (1) r and d are integers, and (2) $(rd) \mid n$.*

In the sequel we shall assume that for the given δ and ϵ, the two conditions of Lemma 2 hold simultaneously. If in this case we can construct a (q, δ, ϵ)-LDC with some length m', it will follow that for any δ and ϵ we can construct a $(q, \Theta(\delta), \Theta(\epsilon))$-LDC with length $\Theta(m')$.

Proof strategy: We first construct an auxiliary function $f : \{0,1\}^{n'} \to \{0,1\}^{\ell}$, where $\ell = \tau m(n'/r)$. The auxiliary function coincides with our encoding function $C' : \{0,1\}^n \to \{0,1\}^{m'(n)}$ when $d = 1$. When $d > 1$, then C' will consist of d applications of the auxiliary function, each on a separate group of n' coordinates of the message x. Recall that $d > 1$ iff $c\epsilon \geq \delta$, and in this case we effectively allow δ to grow while reducing n (see Section 1 for discussion). We will thus have $m'(n) = d\tau m(n'/r)$. We then describe algorithms $\mathsf{Encode}(x)$ and $\mathsf{Decode}^y(k)$ associated with C'. Finally, we show that C' is a (q, δ, ϵ)-LDC with length $m'(n)$. Note that we have ensured r, d, and $n'/r = n/(dr)$ are all integers.

An auxiliary function: Let $\mu_1, \ldots, \mu_\tau$ be the set of strings in $\{0,1\}^r$ guaranteed by Lemma 1. For each $i \in [\tau]$, let s_i be the concatenation of n'/r copies of μ_i. For each $j \in [n'/r]$, let B_j be the set $B_j = \{(j-1)r+1,\ (j-1)r+2,\ \ldots,\ jr\}$. The B_j partition the interval $[1, n']$ into n'/r contiguous blocks each of size r. We now explain how to compute the auxiliary function $f(u)$ for $u \in \{0,1\}^{n'}$. Compute $w_1 = u \oplus s_1, w_2 = u \oplus s_2, \ldots, w_\tau = u \oplus s_\tau$. For each $i \in [\tau]$, compute $z_i \in \{0,1\}^{n'/r}$ as follows: $\forall j \in [n'/r], \quad z_{i,j} = \text{majority}(w_{i,k} \mid k \in B_j)$. Then $f(u)$ is defined to be, $f(u) = C(z_1) \circ C(z_2) \cdots \circ C(z_\tau)$, where $\circ$ denotes string concatenation. Observe that $|f(u)| = \tau m(n'/r)$.

The LDC: We describe the algorithm $\mathsf{Encode}(x)$ associated with our encoding $C' : \{0,1\}^n \to \{0,1\}^{m'(n)}$. We first partition x into d contiguous substrings $u_1, \ldots, u_d$, each of length n'. Then, $\mathsf{Encode}(x) = C'(x) = f(u_1) \circ f(u_2) \cdots \circ f(u_d)$. Observe that $|C'(x)| = m'(n) = d\tau m(n'/r)$. Next we describe the algorithm $\mathsf{Decode}^y(k)$. We think of y as being decomposed into $y = y_1 \circ y_2 \cdots \circ y_d$, where each y_h, $h \in [d]$, is a block of $m'(n)/d = \tau m(n'/r)$ consecutive bits of y. Let h be such that x_k occurs in u_h. Further, we think of y_h as being decomposed into $y_h = v_1 \circ v_2 \cdots \circ v_\tau$, where each v_i, $i \in [\tau]$, is a block of $m(n'/r)$ consecutive bits of y_h.

To decode, first choose a random integer $i \in [\tau]$. Next, let $j \in [n'/r]$ be such that $(k \bmod d) + 1 \in B_j$. Simulate the decoding algorithm $A^{v_i}(j)$ associated with C. Suppose the output of $A^{v_i}(j)$ is the bit b. If the kth bit of s_i is 0, output b, else output $1 - b$. The following is our main theorem.

Theorem 6. *Given a family of $(q, \delta, 1/2 - \beta\delta)$-LDCs of length $m(n)$, where $\beta > 0$ is any constant, and $\delta < 1/(2\beta)$ is arbitrary (i.e., for a given n, the same encoding function C is a $(q, \delta, 1/2 - \beta\delta)$-LDC for any $\delta < 1/(2\beta)$), there is a family of non-linear $(q, \Theta(\delta), \epsilon)$-LDCs of length $O(dr^2)m(n'/r)$ for any $\delta, \epsilon \in [\Omega(n^{-1/2}),\ O(1)]$, where $d = \max(1, O(\epsilon/\delta))$, $r = O((\epsilon + \delta)^{-2})$, and $n' = n/d$.*

Proof. We show that C' is a (q, δ, ϵ)-LDC with length $m'(n) = d\tau m(n'/r)$.

First, observe that $\mathsf{Decode}^y(k)$ always makes at most q queries since the decoder A of C always makes at most q queries. Also, we have already observed that $|C'(x)| = m'(n) = d\tau m(n'/r)$. Now, let $x \in \{0,1\}^n$ and $k \in [n]$ be arbitrary. Let h be such that x_k occurs in u_h.

First, consider the case that $c\epsilon < \delta$, so that $h = d = 1$. Suppose k occurs in the set B_j. By Theorem 5 and the definition of r, for at least a $\frac{1}{2} + \frac{c}{r^{1/2}} = \frac{1}{2} + (1 + 2\beta c)\epsilon + 2\beta\delta$ fraction of the τ different z_i, we have $z_{i,j} = y_{i,k} = x_k \oplus s_{i,k}$. Since i is chosen at random by $\mathsf{Decode}^y(k)$, we have

$\Pr_i[z_{i,j} = x_k \oplus s_{i,k}] > \frac{1}{2} + (1 + 2\beta c)\epsilon + 2\beta\delta$. In case that $z_{i,j} = x_k \oplus s_{i,k}$, we say i is *good*. Let $\mathcal{E}$ be the event that the i chosen by the decoder is good, and let G be the number of good i. We think of the received word $y = y_1$ (recall that $d = 1$) as being decomposed into $y = v_1 \circ v_2 \cdots \circ v_\tau$. The adversary can corrupt a set of at most $\delta m'(n)$ positions in $C'(x)$. Suppose the adversary corrupts $\delta_i m'(n)$ positions in $C(z_i)$, that is, $\Delta(C(z_i), v_i) \leq \delta_i m'(n)$. So we have the constraint $0 \leq \frac{1}{\tau}\sum_i \delta_i \leq \delta$.

Conditioned on $\mathcal{E}$, the decoder recovers $z_{i,j}$ with probability at least $\frac{1}{G}\sum_{\text{ good } i}(1 - \beta\delta_i) = 1 - \frac{\beta}{G}\sum_{\text{ good } i}\delta_i \geq 1 - \frac{\tau\beta\delta}{G} \geq 1 - 2\beta\delta$, where we have used that $G \geq \tau/2$. In this case the decoder recovers x_k by adding $s_{i,k}$ to $z_{i,j}$ modulo 2. Thus, the decoding probability is at least $\Pr[\mathcal{E}] - 2\beta\delta \geq \frac{1}{2} + (1 + 2\beta c)\epsilon + 2\beta\delta - 2\beta\delta > \frac{1}{2} + \epsilon$. Now consider the case that $c\epsilon \geq \delta$, so that d may be greater than 1. The number of errors in the substring $f(u_h)$ of $C'(x)$ is at most $\delta m'(n) = \delta d\tau m(n'/r) = \delta(c\epsilon/\delta)\tau m(n'/r) = c\epsilon|f(u_h)|$, so there is at most a $c\epsilon$ fraction of errors in the substring $f(u_h)$. Again supposing that $(k \bmod d) + 1 \in B_j$, by Theorem 5 we deduce that $\Pr_i[z_{i,j} = x_k \oplus s_{i,k}] > \frac{1}{2} + (1 + 2\beta c)\epsilon + 2\beta\delta$. We define a good i and the event $\mathcal{E}$ as before. We also decompose y_h into $y_h = v_1 \circ v_2 \cdots \circ v_\tau$. By an argument analogous to the case $d = 1$, the decoding probability is at least $\Pr[\mathcal{E}] - 2\beta c\epsilon > \frac{1}{2} + (1 + 2\beta c)\epsilon + 2\beta\delta - 2\beta c\epsilon > \frac{1}{2} + \epsilon$, as needed.

We defer the proofs of the next two corollaries to the full version, which follow by plugging in Hadamard's and Yekhanin's codes into Theorem 6.

Corollary 1. *For any* $\delta, \epsilon \in [\Omega(n^{-1/2}),\ O(1)]$*, there is a* $(2, \delta, \epsilon)$*-LDC of length* $m = \text{poly}(\delta^{-1}, \epsilon^{-1}) \exp(\max(\delta, \epsilon)\delta n)$.

Corollary 2. *For any* $\delta, \epsilon \in [\Omega(n^{-1/2}),\ O(1)]$ *and any prime of the form* $2^t - 1$*, there is a* $(3, \delta, \epsilon)$*-LDC with* $m = \text{poly}(\delta^{-1}, \epsilon^{-1}) \exp\left((\max(\delta, \epsilon)\delta n)^{1/t}\right)$.

4 The Lower Bound

Consider a (q, δ, ϵ)-LDC C with length m which has a decoder that has n matchings $M_1, \ldots, M_n$ of edges on the complete q-uniform hypergraph, whose vertices are identified with positions of the codeword. On input $i \in [n]$ and received word y, the decoder chooses $e = \{a_1, \ldots, a_q\} \in M_i$ uniformly at random and outputs $\bigoplus_{j=1}^{q} y_{a_j}$. All known LDCs, including our non-linear LDCs, satisfy this property. In this case we say that C has a *matching sum decoder*.

Any linear $(2, \delta, \epsilon)$-LDC C can be transformed into an LDC with slightly worse parameters, but with the same encoding function and a matching sum decoder. Indeed, identify the m positions of the encoding of C with linear forms v, where $C(x)_v = \langle x, v\rangle$. Obata [6] has shown that such LDCs have matchings M_i of edges $\{u, v\}$ with $u \oplus v = e_i$, where $|M_i| \geq \beta\delta m$ for a constant $\beta > 0$. By replacing δ with $\delta' = \beta\delta/3$, the decoder can query a uniformly random edge in M_i and output the correct answer with probability at least $(\beta\delta m - \beta\delta m/3)/(\beta\delta m) \geq 2/3$. One can extend this to linear LDCs with $q > 2$ by generalizing Obata's argument.

Theorem 7. *Any* $(2, \delta, \epsilon)$*-LDC* C *with a matching sum decoder satisfies* $m \geq \exp(\max(\delta, \epsilon)\delta n)$.

Proof. For each $i \in [n]$, let the matching M_i of the matching sum decoder satisfy $|M_i| = c_i m$. We may assume, by relabeling indices, that $c_1 \leq c_2 \leq \cdots \leq c_n$. Let $\bar{c} = \sum_i c_i/n$ be the average of the c_i. For each edge $e = \{a, b\} \in M_i$, let $p_{i,e}$ be the probability that $C(x)_a \oplus C(x)_b$ equals x_i for a uniformly chosen $x \in \{0,1\}^n$. The probability, over a random $x \in \{0,1\}^n$, that the decoder outputs x_i if there are no errors is $\psi_i = \sum_{e \in M_i} p_{i,e}/|M_i|$, which is at least $1/2+\epsilon$. But ψ_i is also at least $1/2+\delta/c_i$. Indeed, otherwise there is a fixed x for which it is less than $1/2 + \delta/c_i$. For this x, say $e = \{a, b\}$ is *good* if $C(x)_a \oplus C(x)_b = x_i$. Then $\sum_{\text{good } e \in M_i} 1/|M_i| < 1/2+\delta/c_i$. By flipping the value of exactly one endpoint of δm good $e \in M_i$, this probability drops to $1/2$, a contradiction.

We first transform the LDC C to another code C'. Identify the coordinates of x with indices $0, 1, \ldots, n-1$. For $j = 0, \ldots, n-1$, let π_j be the j-th cyclic shift of $0, \ldots, n-1$, so for $x = (x_0, \ldots, x_{n-1}) \in \{0,1\}^n$, we have that $\pi_j(x) = (x_j, x_{j+1}, \ldots, x_{j-1})$. We define $C'(x) = C(\pi_0(x)) \circ C(\pi_1(x)) \cdots \circ C(\pi_{n-1}(x))$. Then $m' = |C'(x)| = n|C(x)|$. For $j, k \in \{0, 1, \ldots, n-1\}$, let $M_{j,k}$ be the matching M_k in the code $C(\pi_j(x))$. Define the n matchings $M'_0, \ldots, M'_{n-1}$ with $M'_i = \cup_{j=0}^{n-1} M_{j,i-j}$.

We need another transformation from C' to a code C''. For each $i \in \{0, \ldots, n-1\}$, impose a total ordering on the edges in M'_i by ordering the edges $e_1, \ldots, e_{|M'_i|}$ so that $p_{i,e_1} \geq p_{i,e_2} \cdots \geq p_{i,e_{|M'_i|}}$. Put $t = \lfloor 1/(2\bar{c}) \rfloor$, and let C'' be the code with entries indexed by ordered multisets S of $[m']$ of size t, where $C''_S(x) = \bigoplus_{v \in S} C'(x)_v$. Thus, $m'' = |C''(x)| = (m')^t$. Consider a random entry $S = \{v_1, \ldots, v_t\}$ of C''. Fix an $i \in \{0, 1, \ldots n-1\}$. Say *$S$ hits i* if $S \cap \left(\cup_{e \in M'_i} e\right) \neq \emptyset$. Now, $|\cup_{e \in M'_i} e| = 2|M'_i| = 2\bar{c}m'$, so, $\Pr[S \text{ hits } i] \geq 1 - (1-2\bar{c})^t \geq 1 - e^{-2\bar{c}t} \geq 1 - e^{-1} > 1/2$. Thus, at least a $1/2$ fraction of entries of C'' hit i. We can group these entries into a matching M''_i of edges of $[m'']$ with $|M''_i| \geq m''/4$ as follows. Consider an S that hits i and let $e = \{a, b\}$ be the *smallest* edge of

M_i' for which $S \cap \{a, b\} \neq \emptyset$, under the total ordering of edges in M_i' introduced above. Since S is ordered, we may look at the *smallest* position j containing an entry of e. Suppose, w.l.o.g., that $S_j = a$. Consider the ordered multiset T formed by replacing the j-th entry of S with b. Then, $C_S''(x) \oplus C_T''(x) = \bigoplus_{v \in S} C'(x)_v \oplus \bigoplus_{v \in T} C'(x)_v = 2 \bigoplus_{v \notin e} C'(x)_v \oplus (C'(x)_a \oplus C'(x)_b) = C'(x)_a \oplus C'(x)_b$. Given T, the smallest edge hit by T is e, and this also occurs in position j. So the matching M_i'' is well-defined and of size at least $m''/4$.

We will also need a more refined statement about the edges in M_i''. For a random entry S of C'', say S *hits* i *by time* j if $S \cap \left(\cup_{\ell=1}^{j} \cup_{e \in M_{\ell, i-\ell}} e \right) \neq \emptyset$. Let $\sigma_j = \sum_{\ell=1}^{j} c_\ell$. Now, $| \cup_{\ell=1}^{j} \cup_{e \in M_{\ell, i-\ell}} e| = 2\sigma_j m = 2\sigma_j m'/n$. Thus,

$$\Pr[S \text{ hits } i \text{ by time } j] \geq 1 - \left(1 - \frac{2\sigma_j}{n}\right)^t \geq 1 - e^{-\frac{2\sigma_j t}{n}} \geq 1 - e^{-\frac{\sigma_j}{n\bar{c}}} \geq \frac{\frac{\sigma_j}{n\bar{c}}}{1 + \frac{\sigma_j}{n\bar{c}}},$$

where the last inequality is $1 - e^{-x} > x/(x+1)$, which holds for $x > -1$. Now, $\sigma_j/(n\bar{c}) = \sigma_j / \sum_{\ell=1}^{n} c_\ell \leq 1$, so $\Pr[S \text{ hits } i \text{ by time } j] \geq \sigma_j/(2n\bar{c})$.

For $\{S, T\} \in M_i''$, let $p_{i,\{S,T\}}''$ be the probability over a random x that $C''(x)_S \oplus C''(x)_T = x_i$. Then $p_{i,\{S,T\}}'' = p_{i,e}$, where e is the smallest edge of M_i' hit by S and T. We define $\psi_i'' = \frac{1}{|M_i''|} \sum_{\{S,T\} \in M_i''} p_{i,\{S,T\}}''$, which is the probability that the matching sum decoder associated with C'' with matchings M_i'' outputs x_i correctly for a random x, given that there are no errors in the received word. Let $\phi_{i,j}$ be the probability that the smallest edge $e \in M_i'$ hit by a randomly chosen edge in M_i'' is in $M_{j,i-j}$. Due to our choice of total ordering (namely, within a given $M_{j,i-j}$, edges with larger $p_{j,e}$ value are at least as likely to occur as those with smaller $p_{j,e}$ for a randomly chosen edge in M_i'', conditioned on the edge being in $M_{j,i-j}$), $\psi_i'' \geq \sum_j \phi_{i,j} \psi_j \geq \sum_j \phi_{i,j} \left(\frac{1}{2} + \max(\epsilon, \delta/c_j) \right) = \frac{1}{2} + \sum_j \phi_{i,j} \max(\epsilon, \delta/c_j)$. Observe that $\sum_{\ell=1}^{j} \phi_{i,\ell} \geq \sigma_j/(2n\bar{c})$, and since the expression $\max(\epsilon, \delta/c_j)$ is non-increasing with j, the above lower bound on ψ_i'' can be further lower bounded by setting $\sum_{\ell=1}^{j} \phi_{i,\ell} = \sigma_j/(2n\bar{c})$ for all j. Then $\phi_{i,j}$ is set to $c_j/(2n\bar{c})$ for all j, and we have $\psi_i'' \geq 1/2 + \max(\epsilon, \delta/\bar{c})/2$.

Let $\bar{r} = \max(\epsilon, \delta/\bar{c})/2$. We use quantum information theory to lower bound m''. For each $j \in [m'']$, replace the j-th entry of $C''(x)$ with $(-1)^{C''(x)_j}$. We can represent $C''(x)$ as a vector in a state space of $\log m''$ qubits $|j\rangle$. The vector space it lies in has dimension m'', and its standard basis consists of all vectors $|b\rangle$, where $b \in \{0, 1\}^{\log m''}$ (we can assume m'' is a power of 2). Define $\rho_x = \frac{1}{m''} C(x)^\dagger C(x)$. It is easy to verify that ρ_x is a density matrix. Consider the $n + \log m''$ qubit quantum system

XW: $\frac{1}{2^n}\sum_x |x\rangle\langle x| \otimes \rho_x$. We use X to denote the first system, X_i for its qubits, and W for the second subsystem. By Theorem 11.8.4 of [15], $S(XW) = S(X) + \frac{1}{2^n}\sum_x S(\rho_x) \geq S(X) = n$. Since W has $\log m''$ qubits, $S(W) \leq \log m''$, hence $S(X : W) = S(X) + S(W) - S(XW) \leq S(W) \leq \log m''$. Using a chain rule for relative entropy and a highly non-trivial inequality known as the strong subadditivity of the von Neumann entropy, we get $S(X \mid W) = \sum_{i=1}^n S(X_i \mid X_1, \ldots, X_{i-1}, W) \leq \sum_{i=1}^n S(X_i \mid W)$. In the full version, we show that $S(X_i \mid W) \leq H(\frac{1}{2} + \frac{\bar{r}}{2})$. That theorem is a generalization of the analogous theorem of [8], as here we just have matchings M_i'' for which the average probability that the sum of endpoints of an edge in M_i'' is at least $\frac{1}{2} + \bar{r}$, whereas in [8] this was a worst case probability. Putting everything together, $n - \sum_{i=1}^n H\left(\frac{1}{2} + \frac{\bar{r}}{2}\right) \leq S(X) - \sum_{i=1}^n S(X_i \mid W) \leq S(X) - S(X \mid W) = S(X : W) \leq \log m''$. Now, $H(\frac{1}{2} + \frac{\bar{r}}{2}) = 1 - \Omega(\bar{r}^2)$, and so $\log m'' = \Omega(n\bar{r}^2)$. But $\log m'' = O(t)\log m' = O(t)\log nm = O(t \log m) = O\left(\frac{1}{\bar{c}}\log m\right)$. Thus, $m \geq \exp\left(n\bar{c}\bar{r}^2\right)$. If $\delta \geq \epsilon$, then $\delta/\bar{c} \geq \epsilon$, and so $\bar{r} \geq \delta/\bar{c}$. Thus, $\bar{c}\bar{r}^2 \geq \delta^2/\bar{c} \geq \delta^2$. Otherwise, $\epsilon > \delta$, and so $\bar{c}\bar{r}^2 \geq \max(\bar{c}\epsilon^2, \delta^2/\bar{c})$, which is minimized if $\bar{c} = \delta/\epsilon$ and equals $\epsilon\delta$. Thus, $m \geq \exp\left(\max(\delta, \epsilon)\delta n\right)$.

5 A Better Upper Bound for Large δ

We improve the dependence on δ of 3-query LDCs, while only increasing m by a constant factor in the exponent. The proof uses a similar technique to that used for constructing the auxiliary code C'' in the previous section.

Theorem 8. *For any $\delta > 0$ and any constant $\eta > 0$, there is a linear $(3, \delta, 1/2 - 3\delta - \eta)$-LDC with $m = \exp\left(n^{1/t}\right)$ for any prime $2^t - 1$.*

Proof. Let $\gamma > 0$ be a constant to be determined, which will depend on η. Let C be the linear $(3, \delta, 1/2 - 6\delta)$-LDC with $m = \exp\left(n^{1/t}\right)$ constructed in [9]. The LDC C has a matching sum decoder by definition [9]. We identify the positions of C with linear forms $v_1, \ldots, v_m$. We first increase the length of C - for each $j \in [m]$, we append to C both a duplicate copy of v_j, denoted a_j, and a copy of the zero function, denoted b_j. Thus, a_j computes $\langle v_j, x\rangle$ and b_j computes $\langle 0, x\rangle = 0$. Notice that the resulting code C' is a $(3, \delta/3, 1/2 - 6\delta)$-LDC with length $m' = 3m$, and that C' has a matching Z of m triples $\{v_j, a_j, b_j\}$ with $v_j \oplus a_j \oplus b_j = 0$. For each triple $\{v_j, a_j, b_j\}$, we think of it as a *directed cycle* with edges $(v_j, a_j), (a_j, b_j), (b_j, v_j)$. For any $\delta > 0$, the LDC C also has n matchings $M_1, \ldots, M_n$ of triples of $v_1, \ldots, v_m$ so that for all $i \in [n]$ and all

$e = \{v_a, v_b, v_c\} \in M_i$, we have $v_a \oplus v_b \oplus v_c = e_i$, where e_i is the i-th unit vector. We prove the following property of C in the full version.

Lemma 3. *For all $i \in [n]$, $|M_i| \geq m/18$.*

Now, for each $i \in [n]$ and for each triple $\{a, b, c\} \in M_i$, we think of the triple as a directed cycle with edges $(a, b), (b, c), (c, a)$ for some arbitrary ordering of a, b, and c. Define the parameter $p = \lceil 18 \ln 1/(3\gamma) \rceil$. We form a new linear code C'' indexed by all ordered multisets $S \subset [m']$ of size p. Let $m'' = |C''(x)| = (m')^p$. We set the entry $C''_S(x)$ equal to $\bigoplus_{v \in S} C'_v(x)$. For $i \in [n]$, arbitrarily impose a total order $\succeq$ on the triples in M_i. For a particular ordered multiset S_1, we say that S_1 *hits* M_i if there is a triple $e \in M_i$ for which $e \cap S_1 \neq \emptyset$. Then, $\Pr[S_1 \text{ hits } M_i] \geq 1 - \left(1 - \frac{3|M_i|}{m'}\right)^p \geq 1 - \left(1 - \frac{1}{18}\right)^p \geq 1 - e^{-\frac{p}{18}} \geq 1 - 3\gamma$. For any S_1 that hits M_i, let $\{a, b, c\}$ be the smallest triple hit, under the total ordering $\succeq$. Since S_1 is ordered, we may choose the smallest of the p positions in S_1 which is in $\{a, b, c\}$. Let j be this position. Suppose the j-th position contains the linear form a, and that $(a, b), (b, c)$, and (c, a) are the edges of the directed cycle associated with $\{a, b, c\}$. Consider the triple $\{S_1, S_2, S_3\}$ formed as follows.

```
Triple-Generation($S_1$):

1. Set the $j$-th position of $S_2$ to $b$, and the $j$-th position of $S_3$ to $c$.
2. For all positions $k \neq j$, do the following,
   (a) If $v_\ell$ is in the $k$-th position of $S_1$, then put $a_\ell$ in the $k$-th position
       of $S_2$ and $b_\ell$ in the $k$-th position of $S_3$.
   (b) If $a_\ell$ is in the $k$-th position of $S_1$, then put $b_\ell$ in the $k$-th position
       of $S_2$ and $v_\ell$ in the $k$-th position of $S_3$.
   (c) If $b_\ell$ is in the $k$-th position of $S_1$, then put $v_\ell$ in the $k$-th position
       of $S_2$ and $a_\ell$ in the $k$-th position of $S_3$.
3. Output $\{S_1, S_2, S_3\}$.
```

Since $v_j \oplus a_j \oplus b_j = 0$ for all j, we have, $(\bigoplus_{v \in S_1} v) \oplus (\bigoplus_{v \in S_2} v) \oplus (\bigoplus_{v \in S_3} v) = a \oplus b \oplus c = e_i$. The elaborate way of generating S_2 and S_3 was done to ensure that, had we computed Triple-Generation(S_2) or Triple-Generation(S_3), we would also have obtained $\{S_1, S_2, S_3\}$ as the output. This is true since, independently for each coordinate, we walk along a directed cycle of length 3. Thus, we may partition the ordered sets that hit M_i into a matching M''_i of $m''/3 - \gamma m''$ triples $\{S_1, S_2, S_3\}$ containing linear forms that sum to e_i.

Consider the following decoder for C'': on input $i \in [n]$ with oracle access to y, choose a triple $\{S_1, S_2, S_3\} \in M_i''$ uniformly at random and output $y_{S_1} \oplus y_{S_2} \oplus y_{S_3}$. If the adversary corrupts at most $\delta m''$ positions of C'', then at most $\delta m''$ triples in M_i'' have been corrupted, and so the recovery probability of the decoder is at least $\frac{|M_i''|-\delta m''}{|M_i''|} = \frac{\frac{m''}{3}-\gamma m''-\delta m''}{\frac{m''}{3}-\gamma m''} = 1 - \frac{3\delta}{1-3\gamma} \geq 1 - 3\delta - \eta$, where the final inequality follows for a sufficiently small constant $\gamma > 0$. So C'' is a $(3, \delta, 1/2 - 3\delta - \eta)$-LDC. The length of C'' is $m'' = (3m)^p = m^{O(1)} = \exp\left(n^{1/t}\right)$. This completes the proof.

Acknowledgment: The author thanks T.S. Jayram and the anonymous referees for many helpful comments.

References

1. Sipser, M., Spielman, D.A.: Expander codes. IEEE Trans. Inform. Theory, 42:1710-1722 (1996)
2. Katz, J., Trevisan, L.: On the efficiency of local decoding procedures for error-correcting codes. In: STOC. (2000)
3. Trevisan, L.: Some applications of coding theory in computational complexity. Quaderni di Matematica 13:347-424 (2004)
4. Dvir, Z., Shpilka, A.: Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits. SIAM J. Comput. **36**(5) (2007) 1404–1434
5. Goldreich, O., Karloff, H.J., Schulman, L.J., Trevisan, L.: Lower bounds for linear locally decodable codes and private information retrieval. Computational Complexity **15**(3) (2006) 263–296
6. Obata, K.: Optimal lower bounds for 2-query locally decodable linear codes. In: RANDOM. (2002) 39–50
7. Shiowattana, D., Lokam, S.V.: An optimal lower bound for 2-query locally decodable linear codes. Inf. Process. Lett. **97**(6) (2006) 244–250
8. Kerenidis, I., de Wolf, R.: Exponential lower bound for 2-query locally decodable codes via a quantum argument. J. Comput. Syst. Sci. **69**(3) (2004) 395–420
9. Yekhanin, S.: Towards 3-query locally decodable codes of subexponential length. J. ACM **55**(1) (2008)
10. Beimel, A., Ishai, Y., Kushilevitz, E., Raymond, J.F.: Breaking the $O(n^{\frac{1}{2k-1}})$ barrier for information-theoretic private information retrieval. In: FOCS. (2002)
11. Chor, B., Goldreich, O., Hästad, J., Friedman, J., Rudich, S., Smolensky, R.: The bit extraction problem of t-resilient functions. In: FOCS. (1985) 396–407
12. Bennett, C.H., Brassard, G., Robert, J.M.: Privacy amplification by public discussion. SIAM J. Comput **17(2)** (1988) 210–229
13. Stinson, D.R., Massey, J.L.: An infinite class of counterexamples to a conjecture concerning nonlinear resilient functions. J. Cryptology **8**(3) (1995) 167–173
14. Beimel, A., Ishai, Y.: On the power of nonlinear secrect-sharing. In: IEEE Conference on Computational Complexity. (2001) 188–202
15. Nielsen, M.A., Chuang, I.: Quantum computation and quantum information. Cambridge University Press (2000)