3-Query Locally Decodable Codes of Subexponential Length

Klim Efremenko [*]

November 13, 2008

Abstract

Locally Decodable Codes (LDC) allow one to decode any particular symbol of the input message by making a constant number of queries to a codeword, even if a constant fraction of the codeword is damaged. In a recent work [Yek08] Yekhanin constructs a 3-query LDC with sub-exponential length of size $\exp(\exp(O(\frac{\log n}{\log\log n})))$. However, this construction requires a conjecture that there are infinitely many Mersenne primes. In this paper we give an unconditional 3-query LDC construction with a shorter codeword length of $\exp(\exp(O(\sqrt{\log n \log\log n})))$. We also give a 2^r-query LDC with length of $\exp(\exp(O(\sqrt[r]{\log n \log\log^{r-1} n})))$. The main ingredient in our construction is the existence of super-polynomial size set-systems with restricted intersections by [Gro00] which hold only over composite numbers.

1 Introduction

Locally decodable codes (LDCs) are codes that allow to retrieve any symbol of the original message by reading only a constant number of symbols from the codeword. Formally a code C is said to be locally decodable with parameters (q, δ, ε) if it is possible to recover any bit x_i of message x by making at most q queries to $C(x)$. Such that if up to a δ fraction of $C(x)$ is corrupted then the decoding algorithm will return the correct answer with probability at least $1 - \varepsilon$.

Locally decodable codes have many applications in cryptography and complexity theory, see surveys in [Tre04] and [Gas04]. The first formal definition of locally decodable codes was given by Katz and Trevisan in [KT00]. The Hadamard code is the most famous 2-query locally decodable code of length 2^n. For a two-query LDC tight lower bounds of $2^{\theta(n)}$ were given for linear codes in [GKST02] and [KdW03] proved tight lower bounds for two queries for arbitrary codes. For an arbitrary number of queries Katz and Trevisan [KT00] established super-linear lower bounds of $\Omega(n^{q/(q-1)})$ for LDCs with q queries. This lower bound was later improved in [KdW03] to $\Omega\left((\frac{n}{\log n})^{1+1/(\lceil q/2\rceil-1)}\right)$ and in [Woo07] to $\Omega\left(\frac{n^{1+1/(\lceil q/2\rceil-1)}}{\log n}\right)$.

[*] Weizmann Institute of Science, Rehovot 76100, Israel, Bar-Ilan University, 52900 Ramat-Gan, Israel; klimefrem@gmail.com

For many years it was conjectured that LDCs should have an exponential dependence on n for any constant number of queries, until Yekhanin's recent breakthrough [Yek08]. Yekhanin obtained 3-query LDCs with sub-exponential length of $\exp(\exp(O(\frac{\log n}{\log\log n})))$ under a highly believable conjecture that there are infinitely many Mersenne primes. Using the known Mersenne primes, Yekhanin also obtained unconditional results which significantly improved the previous results on LDCs(i.e. length of $\exp(n^{10^{-7}})$). In [KY08] Kedlaya and Yekhanin proved that infinitely many Mersenne numbers with large prime factors are essential for Yekhanin's construction.

Our Results In this paper we give an unconditional construction of 3-query LDC with sub-exponential codeword length. The length that we achieve for 3 queries is:

$$\exp\exp(O(\sqrt{\log n \log\log n})).$$

We also give a 2^r-query LDC with a codeword length $\exp\exp(O(\sqrt[r]{\log n \log\log^{r-1} n}))$.

Our construction is a kind of a generalization and simplification of [Yek08]. We extend Yekhanin's construction to work not only with primes but also with composite numbers. Raghavendra in [Rag07] gives a nice presentation of Yekhanin's construction using homomorphisms, and we will follow this approach. The main ingredient in our construction is the existence of super-polynomial size set-systems with restricted intersections [Gro00], which hold only over composite numbers.

Private Information Retrieval schemes: The notion of locally decodabale codes is closely related to the notion of private information retrieval(PIR) schemes. PIR schemes with k servers is a protocol which allows for a user to access a database distributed between k servers without yielding any information on the identity of the accessed place to any individual server (we assume that there is no communication between servers). The main parameter of interest in PIR schemes is the total communication complexity between the user and the servers. PIR schemes were first introduced by [CGKS95]. After that there were many works written on this topic, see [CGKS95, Amb97, Man98, Ito99, BIK05, GKST06, KdW03, RY07, WdW05, Yek08]. The best upper bound for 2-server PIR is $O(n^{1/3})$ due to [CGKS95]. The best upper bound of 3 and more server PIR schemes is $\exp\left(O\left(\frac{\log n}{\log^{1-\varepsilon}\log n}\right)\right)$ due to [Yek08] which is based on the construction of LDCs.

Let us define formally perfect PIR schemes:

Definition 1.1. A one-round perfect *private information retrieval* scheme is a randomized algorithm $\mathcal{U}$ (for the user), and k deterministic algorithms $\mathcal{S}_1, \ldots \mathcal{S}_k$ (for the servers), s.t.

1. (a) On input $i \in [n]$ the user $\mathcal{U}$ produces k random queries $q_1 \ldots q_k$ and sends them to respective servers.
   (b) Each server based on his query q_j and database $\mathcal{D}$ produces a response $r_j = \mathcal{S}_i(\mathcal{D}, q_j)$ and sends it back to the user.
   (c) The user based on $i, r_1, \ldots, r_k$ and his randomness calculates $\mathcal{D}[i]$.
2. The distribution of each query q_j is independent of the input i.

The communication complexity of this protocol is a total number of bits exchanged between user and servers.

It is well known that LDCs with perfectly smooth decoder imply PIR schemes. In particular, as in [Yek08], our LDC yields a PIR schemes with communication complexity $\exp(O(\sqrt{\log n \log\log n}))$ for 3-servers and $\exp(O(\sqrt[r]{\log n \log^{r-1} \log n}))$ for 2^r-servers.

2 Definitions and Basic Facts

We will use the following standard mathematical notation:

- $[s] = \{1, \dots s\}$;
- $\mathbb{F}_q = GF(q)$ is a finite field of q elements;
- $\mathbb{F}^*$ is a multiplicative group of the field;
- $\mathbb{Z}_m = \mathbb{Z}/m\mathbb{Z}$, the integers modulo m;
- $d_H(x, y)$ denotes the Hamming distance between vectors $x, y \in \Sigma^n$, i.e. number of indices where $x_i \neq y_i$.

Definition 2.1. A code C over a field $\mathbb{F}$, $C : \mathbb{F}^n \mapsto \mathbb{F}^N$ is said to be (q, δ, ε) locally decodable if there exist randomized decoding algorithms d_i for $i = 1, 2, \dots n$ such that for all $i = 1, 2, \dots n$ the following holds:

1. For every message $\vec{x} = (x_1, x_2, \dots x_n) \in \mathbb{F}^n$ and for every $\vec{y} \in \mathbb{F}^N$ such that $d_H(C(\vec{x}), \vec{y}) \leq \delta N$ it holds that $\Pr(d_i(\vec{y}) = x_i) \geq 1 - \varepsilon$; i.e. the decoding algorithm succeeds to recover the relevant symbol even if up to δ fraction of the codeword is damaged.
2. The algorithm d_i makes at most q queries to y.

A code C is called linear if C is a linear transformation over $\mathbb{F}$. A locally decodable code is called nonadaptive if d_i makes all its queries simultaneously. Our constructions of locally decodable codes are linear and nonadaptive.

Definition 2.2. A code C is said to have a *perfectly smooth decoder* if $d_i(C(\vec{x})) = x_i$ for every $\vec{x}$ and each query of d_i is uniformly distributed over $[N]$.

Fact 2.3 (from [Tre04])**.** *Any code with a perfectly smooth decoder which makes q queries is also $(q, \delta, q\delta)$ locally decodable.*

We will use the following fact:

Fact 2.4. *For every odd m there exists a finite field $\mathbb{F} = GF(2^t)$, where $t \leq m$, and an element $\gamma \in \mathbb{F}$ that is a generator of a multiplicative group of size m, i.e. $\gamma^m = 1$ and $\gamma^i \neq 1$ for $i = 1, 2, \dots m - 1$.*

Proof. Since m is odd $2 \in \mathbb{Z}_m^*$. Therefore, there exists $t < m$ such that $2^t \equiv 1 \mod m$. Let us set $\mathbb{F} = GF(2^t)$. The size of the multiplicative group $\mathbb{F}^*$ is $2^t - 1$ and therefore it is divisible by m. Let g be a generator of $\mathbb{F}^*$. Then $\gamma = g^{\frac{2^t-1}{m}}$ is a generator of a multiplicative group of size m. □

In Appendix A for simple construction of S-matching vectors we will need the following definition and fact about tensor product:

Definition 2.5 (Tensor Product)**.** Let R be a ring and let $\vec{x}, \vec{y} \in R^n$. The *tensor product of* $\vec{x}, \vec{y}$ denoted by $\vec{x} \otimes \vec{y} \in R^{n^2}$, is defined by $\vec{x} \otimes \vec{y}(i,j) \triangleq x_i \cdot y_j$, (where we identify $[n^2]$ with $[n] \oplus [n]$.) In the same way we define *the ℓ'th tensor power* $\vec{x}^{\otimes \ell} \in R^{n^\ell}$ by

$$\vec{x}^{\otimes \ell}(i_1, i_2, \dots i_\ell) \triangleq \prod_{j=1}^{\ell} x_{i_j}. \tag{1}$$

We will use only the following fact about tensor products:

Fact 2.6.
$$\langle u^{\otimes \ell}, v^{\otimes \ell} \rangle = \langle u, v \rangle^\ell$$

Proof.

$$\begin{gathered}\langle u^{\otimes \ell}, v^{\otimes \ell} \rangle = \sum_{1 \le i_1, i_2, \dots i_l \le m} \left(\prod_{j=1}^{\ell} u_{i_j} \prod_{j=1}^{\ell} v_{i_j} \right) = \\ \left(\sum_{1 \le i_1 \le m} u_{i_1} v_{i_1} \right) \cdots \left(\sum_{1 \le i_\ell \le m} u_{i_\ell} v_{i_\ell} \right) = \langle u, v \rangle^\ell . \end{gathered} \tag{2}$$

□

3 Locally Decodable Codes

In this construction we follow Yekhanin's general framework. Our construction consists of two parts. The first part is a construction of matching sets of vectors that correspond to “combinatorially nice” sets used in [Yek08]. The second part is a construction of an S-decoding polynomial with a small number of monomials, which correspond to “algebraically nice” sets used in [Yek08]. Let us fix some composite number m for our construction. We wiil give a general scheme for construction of LDCs followed by a concrete example of a 3-query LDC.

3.1 Matching sets of vectors

All inner products $\langle x, y \rangle$ in this section are done $\mod m$.

Definition 3.1. The family of vectors $\{u_i\}_{i=1}^n, u_i \in (\mathbb{Z}_m)^h$ is said to be S-*matching* if the following conditions hold:

1. $\langle u_i, u_i \rangle = 0$ for every $i \in [n]$.
2. $\langle u_i, u_j \rangle \in S$ for every $i \neq j$.

The goal of this subsection is to construct large S-matching family over a small domain. The main advantage of working with composite numbers comes from the following lemma from [Gro00], which holds only for composite numbers.

Lemma 3.2 (Theorems 1.2 and 1.4 from [Gro00]). *Let $m = p_1 p_2 \dots p_r$ be a product of r distinct primes p_i. Then there exists $c = c(m) > 0$, such that for every integer $h > 0$, there exists an explicitly constructible set-system $\mathcal{H}$ over a universe of h elements (i.e $\mathcal{H}$ is a set of subsets of $[h]$) and there is a set $S \subset \mathbb{Z}_m$ such that:*

1. $|\mathcal{H}| \geq \exp(c \frac{(\log h)^r}{\log\log^{r-1} h})$,
2. *Size of every set H in set-system $\mathcal{H}$ is divisible by m i.e. $|H| \equiv 0 \mod m$,*
3. *Let G, H be any two different sets in set system $\mathcal{H}$. Then the size of intersection of G, H modulo m is restricted to be in S. i.e. $\forall G, H \in \mathcal{H}$ such that $G \neq H$. Ut holds that $|G \cap H| \in S \mod m$*
4. *S is a set of size $2^r - 1$ and $0 \notin S$.*
5. *$\forall s \in S$ for all $i = 1, 2, \dots r$ it holds that $s \mod p_i$ is 0 or 1.*

For our construction we will only need the following simple corollary:

Corollary 3.3. *For every h, r and integer $m = p_1 p_2 \dots p_r$ there exists a set S of size $2^r - 1$ and a family of S-matching vectors $\{u_i\}_{i=1}^n, u_i \in (\mathbb{Z}_m)^h$ such that $n \geq \exp(c \frac{(\log h)^r}{\log\log^{r-1} h})$.*

Proof. Let us take set-system $\mathcal{H}$ as in Lemma 3.2. For each set $H \in \mathcal{H}$ we will have one vector $u_H \in (\mathbb{Z}_m)^h$ which is the indicator vector of H. Then it holds that $\langle u_H, u_H \rangle = |H| \equiv 0 \mod m$ and $\langle u_H, u_G \rangle = |H \cap G| \in S \mod m$. □

The construction of [Gro00] is complicated; therefore, we will not give it here. We will give a simple construction of S-matching set in Appendix A which is less strong but it is more simple.

3.2 S-decoding polynomials

Let us fix any odd number m. Recall from Fact 2.4 that there exists t, $\mathbb{F} = GF(2^t)$ and an element $\gamma \in \mathbb{F}$ such that γ is a generator of a multiplicative group of size m. We will first construct a linear code over the field $\mathbb{F}$. In the next section we will show how to reduce the alphabet size to 2.
We will need the following definition:

Definition 3.4. A polynomial $P \in \mathbb{F}[x]$ is called an S-decoding polynomial if the following conditions hold:

- $\forall s \in S \; P(\gamma^s) = 0$,
- $P(\gamma^0) = P(1) = 1$.

Claim 3.1. *For any S such that $0 \notin S$ there exists an S-decoding polynomial P with at most $|S| + 1$ monomials.*

Proof. Let us take $\tilde{P} = \prod_{s \in S}(x - \gamma^s)$. Then $P(x) = \tilde{P}(x)/\tilde{P}(1)$ is an S decoding polynomial. The degree of P is $|S|$. Thus P has at most $|S| + 1$ monomials. □

3.3 The code and its decoding algorithms

Now we are ready to present the construction of our locally decodable codes.
In order to construct our code we will fix some set S and construct S-matching vectors $\{u_i\}_{i=1}^n$, $u_i \in (\mathbb{Z}_m)^h$ and an S-decoding polynomial P. We define a code $C : \mathbb{F}^n \mapsto \mathbb{F}^{m^h}$ where we think of a codeword as a function from $(\mathbb{Z}_m)^h$ to $\mathbb{F}$. Let $e_i \in \mathbb{F}^n$ be the i'th unit vector. We define C by defining $C(e_i)$ for all i. The general definition will follow by the linearity of C, i.e. $C(\sum c_i e_i) \triangleq \sum c_i C(e_i)$. The encoding of e_i is

$$C(e_i) \triangleq (\gamma^{<u_i,x>})_{x \in (\mathbb{Z}_m)^h}. \tag{3}$$

One can think of $C(e_i)$ as a homomorphism from the additive group $(\mathbb{Z}_m)^h$ to the multiplicative group $\mathbb{F}^*$. Equivalently, we can write

$$C((c_1, c_2, \ldots c_n)) \triangleq \sum_{i=1}^{n} c_i f_i, \tag{4}$$

where $f_i(x) \triangleq \gamma^{<u_i,x>}$.

We will now describe how to retrieve the i'th coordinate of the message.

Since P is an S-decoding polynomial and $\{u_i\}$ are S-matching vectors, $\langle u_j, u_i \rangle \in S$ for $i \neq j$, and therefore it follows that $P(\gamma^{<u_i,u_i>}) = 1$ and $P(\gamma^{<u_j,u_i>}) = 0$ for all $i, j \in [n], i \neq j$. Write $P(x) = a_0 + a_1 x^{b_1} + a_2 x^{b_2} \ldots a_{q-1} x^{b_{q-1}}$.
Let us now define the decoding algorithm $d_i(w)$, where w is a codeword with up to δ fraction damaged coordinates.

- Choose $v \in (\mathbb{Z}_m)^h$ at random.
- Query $w(v), w(v + b_1 u_i), \ldots w(v + b_{q-1} u_i)$.
- Output

$$c_i = \gamma^{-<u_i,v>} \left(a_0 w(v) + a_1 w(v + b_1 u_i) \ldots a_{q-1} w(v + b_{q-1} u_i)\right). \tag{5}$$

Algorithm 1: The Decoding Algorithm

Lemma 3.5. *The decoding algorithm d_i is a Perfectly Smooth Decoder.*

Proof. The algorithm d_i chooses v uniformly at random. Each of the queries $v, v+b_1u_i, \dots v+b_{q-1}u_i$ is uniformly distributed. Therefore, in order to prove that d_i is a Perfectly Smooth Decoder it is enough to prove that $d_i(C(x)) = x_i$. Note that d_i is a linear mapping so it is enough to prove that $d_i(C(e_i)) = 1$ and $d_i(C(e_j)) = 0$ for $j \neq i$.

$$d_i(C(e_i)) = (\gamma^{-<u_i,v>})(a_0\gamma^{<u_i,v>} + a_1\gamma^{<u_i,v+b_1u_i>} + \dots + a_{q-1}\gamma^{<u_i,v+b_{q-1}u_i>}).$$

But $\langle u_i, v+cu_i\rangle = \langle u_i, v\rangle + c\langle u_i, u_i\rangle = \langle u_i, v\rangle$. So we have,

$$d_i(C(e_i)) = \gamma^{-<u_i,v>}(a_0\gamma^{<u_i,v>} + a_1\gamma^{<u_i,v>} + \dots + a_{q-1}\gamma^{<u_i,v>}) = \\ = a_0 + a_1 \dots + a_{q-1} = P(1) = 1.$$

Now let us prove that

$$\forall i \neq j \;\; d_i(C(e_j)) = 0.$$

We need to show that

$$a_0\gamma^{<u_i,v>} + a_1\gamma^{<u_i,v+b_1u_j>} + \dots + a_{q-1}\gamma^{<u_i,v+b_{q-1}u_j>} = 0.$$

Recall that $P(\gamma^{<u_i,u_j>}) = 0$. Therefore,

$$\gamma^{<u_i,v>}(a_0 + a_1\gamma^{b_1<u_i,u_j>} + \dots + a_{q-1}\gamma^{b_{q-1}<u_i,u_j>}) = \gamma^{<u_i,v>}P(\gamma^{<u_i,u_j>}) = 0.$$

□

The dimension of the code is n-the number of S-matching vectors. The codeword length is $\left|(\mathbb{Z}_m)^h\right| = m^h$ and the number of queries is equal to the number of monomials of P. An immediate corollary from Corollary 3.3 and Claim 3.1 is that we can choose $n \geq \exp(c\frac{(\log h)^r}{\log\log^{r-1}h})$ and an S-decoding polynomial with less than 2^r monomials. Thus we have the following theorem.

Theorem 3.6. *For any r there exists a $(q, \delta, q\delta)$ locally decodable code $C : F^n \mapsto F^N$, with codeword length $N = \exp(\exp(O(\sqrt[r]{\log n \log\log^{r-1} n})))$ and $q \leq 2^r$. Furthermore, q is the minimal number of monomials of S-decoding polynomial.*

Proof. Let $m = p_1 \dots p_r$ be the product of r primes. Fix $h = \exp\left(\left(O(\sqrt[r]{\log n \log\log^{r-1} n})\right)\right)$. From Corollary 3.3 there exists a set S of size $2^r - 1$ and $n = \exp(c\frac{(\log h)^r}{\log\log^{r-1}h})$ S-matching vectors. Using the construction above we get a code C with codeword length m^h and message length n. Fix m to be a constant. Then $m^h = \exp(O(h))$. Therefore,

$$m^h = \exp(O(h)) = \exp\left(\exp\left(O\left(\sqrt[r]{\log n \log\log^{r-1} n}\right)\right)\right).$$

From Claim 3.1 there exists an S-decoding polynomial with $q \leq 2^r$ monomials. Using this polynomial for our decoding algorithm we get from Lemma 3.5 that C has a Perfectly Smooth Decoder which makes q queries. Thus from Fact 2.3 we have that the code C is a $(q, \delta, q\delta)$-LDC. □

The Claim 3.1 gives a trivial polynomial with 2^r monomials. The natural question is: "Do polynomials exist with less monomials?" The answer is **Yes**. Let us give a concrete example of an S-decoding polynomial with 3 monomials. We found this example by an exhaustive search.

Example 3.7. *Let $m = 511 = 7 \cdot 73$ and let $S = \{1, 365, 147\}$. By Corollary 3.3 there exists S-matching vectors $\{u_i\}_{i=1}^n$, $u_i \in (\mathbb{Z}_m)^h$, where $n \geq \exp(c\frac{(\log h)^2}{\log\log h})$. Set*

$$\mathbb{F} = GF(2^9) = \mathbb{F}_2[\gamma]/(\gamma^9 + \gamma^4 + 1).$$

It can be verified that γ is a generator of $\mathbb{F}^*$ and that the polynomial $P(x) := \gamma^{423} \cdot x^{65} + \gamma^{257} \cdot x^{12} + \gamma^{342}$ is an S decoding polynomial with 3 monomials.

An interesting question is what is the best S-decoding polynomial we can choose for $r > 2$? An immediate corollary from this example and Theorem 3.6 is 3-query LDC.

Theorem 3.8. *There exists a $(3, \delta, 3\delta)$ locally decodable code of length $\exp(\exp(O(\sqrt{\log n \log\log n})))$.*

4 Binary Locally Decodable Codes

In this section we will think of $\mathbb{F}_{2^t}$ as a vector space $\mathbb{F}_2^t$ over $\mathbb{F}_2$. We will view multiplication as a linear transformation i.e. for every $a \in \mathbb{F}_{2^t}$ there exists an n by n matrix M_a over $\mathbb{F}_2$ such that $M_a x = ax$.

Assume now that we have message $(c_1, c_2, \ldots, c_n) \in \mathbb{F}_2^n$. First we will view it as a message in $(\mathbb{F}_{2^t})^n$. Now let $w = C(c_1, c_2, \ldots c_n)$, $w \in (\mathbb{F}_{2^t})^{m^h}$ be an encoding of the message as in the previous section. Next let us extend our codeword to be a concatenation of q identical codewords $w_0 \circ w_1 \circ w_{q-1} = w \circ w \circ \ldots \circ w$. Now we will ask the first query from w_0, the second query from w_1 and so on. Note that this does not harm the probability of correct decoding; it only decreases the rate by a factor q (which is negligible in our parameters). The decoding algorithm from the previous section uses some linear combination over $\mathbb{F}_{2^t}$. We can make this combination to be over $\mathbb{F}_2$. Let $P(x) = a_0 + a_1 x^{b_1} + a_2 x^{b_2} \ldots a_{q-1} x^{b_{q-1}}$ be an S-decoding polynomial. Next let us now set our codeword to be

$$\tilde{w_0} \circ \tilde{w_1} \circ \ldots \circ \tilde{w}_{q-1} \triangleq a_0 w \circ a_1 w \ldots \circ a_{q-1} w,$$

where $w = C(x)$ and $a_i w$ is a coordinate wise scalar multiplication. Recall that from Equation 5 we can decode the i-th symbol c_i using the identity:

$$c_i \gamma^{<u_i, v>} = \tilde{w}_0(v) + \tilde{w}_1(v + b_1 u_i) + \ldots \tilde{w}_{q-1}(v + b_{q-1} u_i).$$

Now let us take some linear functional $L : \mathbb{F}_{2^t} \mapsto \mathbb{F}_2$ and apply it on every coordinate of our codeword. Then

$$L(c_i \gamma^{<u_i, v>}) = L(\tilde{w}_0(v)) + L(\tilde{w}_1(v + b_1 u_i)) + \ldots L(\tilde{w}_{q-1}(v + b_{q-1} u_i)).$$

We want that $L(c_i \gamma^{<u_i, v>}) = c_i$. If $c_i = 0$ then always $L(c_i \gamma^{<u_i, v>}) = L(0) = 0$ but the problem is that if $c_i = 1$ then it may happen that $L(c_i \gamma^{<u_i, v>}) = L(\gamma^{<u_i, v>}) = 0$. In order to solve this problem we will not choose v completely at random; we will choose v at random conditioned on $L(\gamma^{<u_i,v>}) = 1$, but this will hurt the smoothness of the code which in turn affects the probability of correct decoding. In order that it will not hurt this probability too much we need to choose L such that for every $i = 1 \ldots n$ $\Pr_v(L(\gamma^{<u_i,v>}) = 1) \geq 1/2$.

Lemma 4.1. *There exists a linear functional $L : \mathbb{F}_{2^t} \mapsto \mathbb{F}_2$ such that*

$$\forall i \in [n] \Pr_{v \in (\mathbb{Z}_m)^h} (L(\gamma^{<u_i,v>}) = 1) \geq 1/2.$$

Proof. Observe that for random v, $\langle u_i, v\rangle$ is a random number in $\mathbb{Z}_m$,since the gcd of u_i's coordinates is 1. Thus it is enough to find L such that

$$\Pr_{j \in \mathbb{Z}_m} (L(\gamma^j) = 1) \geq 1/2.$$

For a constant j and a random L, $\Pr(L(\gamma^j) = 1) = 1/2$ thus, the expectation of $\Pr_{j\in\mathbb{Z}_m}(L(\gamma^j) = 1)$ is $1/2$ i.e.

$$\mathbf{E}_L(\Pr_{j \in \mathbb{Z}_m} (L(\gamma^j) = 1)) = 1/2.$$

Therefore, there exists an L such that

$$\Pr_{j \in \mathbb{Z}_m} (L(\gamma^j) = 1) \geq 1/2.$$

□

Let us describe the reduction formally.
Choose L such that $\Pr_{j\in\mathbb{Z}_m}(L(\gamma^j) = 1) \geq 1/2$. Since m is constant we can find it by exhaustive search in constant time.

1. Given a message $(c_1, c_2, \ldots c_n)$ encode it , by code from previous section $w = C(c_1, c_2, \ldots, c_n)$.

2. Extend it to
$$\tilde{w} \triangleq \tilde{w}_0 \circ \tilde{w}_1 \circ \ldots \circ \tilde{w}_{q-1} \triangleq a_0 w \circ a_1 w \ldots \circ a_{q-1} w.$$

3. Reduce the alphabet by applying L on every symbol of $\tilde{w}$ and return
$$w_0 \circ w_1 \circ \ldots \circ w_{q-1} \triangleq L(\tilde{w}_0) \circ L(\tilde{w}_1) \circ \ldots \circ L(\tilde{w}_{q-1}).$$

Let us define the decoding algorithm $d_i(w)$:

- Choose $v \in (\mathbb{Z}_m)^h$ at random conditioned on $L(\gamma^{<u_i,v>}) = 1$.
- Query $w_0(v), w_1(v + b_1 u_i), \ldots, w_{q-1}(v + b_{q-1} u_i)$.
- Output $c_i = w_0(v) \oplus w_1(v + b_1 u_i) \ldots \oplus w_{q-1}(v + b_{q-1} u_i)$.

Algorithm 2: Decoding Algorithm

Theorem 4.2. *The binary code C defined above is $(q, \delta, 2q\delta)$ locally decodable.*

Proof. We will prove it in two steps.

First let us prove that if at most δ fraction of the codeword $w = w_0 \circ w_1 \ldots \circ w_{q-1}$ is damaged then we query a damaged place with probability at most $2q\delta$. Let δ_i be a fraction of damaged bits in w_i so $\frac{1}{q}\sum \delta_i = \delta$. We chose L such that v is distributed uniformly among half of all possible values. Therefore, the probability that query i will be damaged is at most $2\delta_i$. So the probability that one of the queries will be damaged is at most $\sum 2\delta_i = 2q\delta$.

Next let us prove that if we query only non-damaged places then we will return a correct answer. As before, by linearity it is enough to prove that $d_i(C(e_i)) = 1$ and $d_i(C(e_j)) = 0$ for $i \neq j$.

$$\begin{gathered} d_i(C(e_i)) = L(a_0\gamma^{<u_i,v>}) \oplus L(a_1\gamma^{<u_i,v+b_1u_i>}) \ldots \oplus L(a_{q-1}\gamma^{<u_i,v+b_{q-1}u_i>}) = \\ = L\left(\sum_{j=0}^{q-1} a_j\gamma^{<u_i,v+b_ju_i>}\right) = L\left(\sum_{j=0}^{q-1} a_j\gamma^{<u_i,v>}\right) = \\ L(P(1)\gamma^{<u_i,v>}) = L(\gamma^{<u_i,v>}) \end{gathered}$$

But we choose v such that $L(\gamma^{<u_i,v>}) = 1$. In the same way we can prove that if $C = C(e_j)$ then $c_i = 0$.

$$\begin{gathered} c_i = L(a_0\gamma^{<u_j,v>}) \oplus L(a_1\gamma^{<u_j,v+b_1u_i>}) \ldots \oplus L(a_{q-1}\gamma^{<u_j,v+b_{q-1}u_i>}) = \\ L\left(\gamma^{<u_j,v>}\sum_{t=0}^{q-1} a_t\gamma^{b_t<u_j,u_i>}\right) = L\left(P(\gamma^{<u_i,u_j>})\gamma^{<u_i,v>}\right) = \\ L(0) = 0. \end{gathered}$$

□

We want to mention here that using techniques from [Woo08] Section 5 we can reduce the probability of error to $(q, \delta, q\delta + \varepsilon)$ for any constant $\varepsilon > 0$.

5 Future work

In this paper we give a general construction of LDCs for any S-matching sets and S-decoding polynomials. Any improvement in size of a set-system with restricted intersections will immediately yield improvement in the rate of LDCs. We hope that this paper will give a motivation for future work on set-systems with restricted intersections. We also believe that it is possible to choose an S-decoding polynomial with less monomials as in Example 3.7.

Acknowledgements

I am indebted to Irit Dinur for many helpful in-depth technical discussions and helping me at all stages of this work. I would also like to thank to Venkatesan Guruswami for directing me to [Gro00] and to Ariel Gabizon, Oded Goldreich, Shachar Lovett, Omer Reingold, David Woodruff, and my wife Rivka for their valuable comments.

References

[Amb97] Andris Ambainis. Upper bound on communication complexity of private information retrieval. In Pierpaolo Degano, Roberto Gorrieri, and Alberto Marchetti-Spaccamela, editors, *ICALP*, volume 1256 of *Lecture Notes in Computer Science*, pages 401–407. Springer, 1997.

[BIK05] Amos Beimel, Yuval Ishai, and Eyal Kushilevitz. General constructions for information-theoretic private information retrieval. *J. Comput. Syst. Sci.*, 71(2):213–247, 2005.

[CGKS95] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. In *FOCS*, pages 41–50, 1995.

[Gas04] William I. Gasarch. A survey on private information retrieval (column: Computational complexity). *Bulletin of the EATCS*, 82:72–107, 2004.

[GKST02] Oded Goldreich, Howard J. Karloff, Leonard J. Schulman, and Luca Trevisan. Lower bounds for linear locally decodable codes and private information retrieval. In *IEEE Conference on Computational Complexity*, pages 175–183, 2002.

[GKST06] Oded Goldreich, Howard J. Karloff, Leonard J. Schulman, and Luca Trevisan. Lower bounds for linear locally decodable codes and private information retrieval. *Computational Complexity*, 15(3):263–296, 2006.

[Gro00] Vince Grolmusz. Superpolynomial size set-systems with restricted intersections mod 6 and explicit ramsey graphs. *Combinatorica*, 20(1):71–86, 2000.

[Ito99] Toshiya Itoh. Efficient private information retrieval. *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences Vol.E82-A No.1 pp.11-20*, 1999.

[KdW03] Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. In *STOC*, pages 106–115. ACM, 2003.

[KT00] Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *STOC*, pages 80–86, 2000.

[KY08] Kiran S. Kedlaya and Sergey Yekhanin. Locally decodable codes from nice subsets of finite fields and prime factors of mersenne numbers. In *IEEE Conference on Computational Complexity*, pages 175–186. IEEE Computer Society, 2008.

[Man98] Eran Mann. Private access to distributed information. In *Master's thesis, Technion - Israel Institute of Technology*, 1998.

[Rag07] Prasad Raghavendra. A note on yekhanin's locally decodable codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 2007.

[RY07] Alexander A. Razborov and Sergey Yekhanin. An omega($^{1/3}$) lower bound for bilinear group based private information retrieval. *Theory of Computing*, 3(1):221–238, 2007.

[Tre04] Luca Trevisan. Some applications of coding theory in computational complexity. *Electronic Colloquium on Computational Complexity (ECCC)*, (043), 2004.

[WdW05] Stephanie Wehner and Ronald de Wolf. Improved lower bounds for locally decodable codes and private information retrieval. In *ICALP*, pages 1424–1436, 2005.

[Woo07] David Woodruff. New lower bounds for general locally decodable codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 2007.

[Woo08] David P. Woodruff. Corruption and recovery-efficient locally decodable codes. In *APPROX-RANDOM*, pages 584–595, 2008.

[Yek08] Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. *J. ACM*, 55(1), 2008.

A A simple construction of S-matching vectors

Lemma A.1. *Let $p_1 < p_2 \ldots < p_r$ be any r primes and $m = p_1 \cdot p_2 \ldots p_r$. Then for every t, there exists a set S of size $2^r - 1$ and a family of S-matching vectors $\{u_i\}_{i=1}^n, u_i \in (\mathbb{Z}_m)^h$ such that $n = \binom{t}{m-1}$ and $h = O(t^{p_r - 1})$.*

Proof. Let us first construct a family of vectors $\{u'_i\}_{i=1}^n, u'_i \in (\mathbb{Z}_m)^{t+1}$ such that:

1. $\langle u'_i, u'_i \rangle = 0$ for $i \in [n]$.
2. $\langle u'_i, u'_j \rangle \neq 0$ for $i \neq j$.

Identify the subsets of $[t] = \{1, 2, \ldots t\}$ of size $m - 1$ with $\{1, \ldots, \binom{t}{m-1}\}$. For every subset $A \subseteq [t]$ of size $m-1$, let $u'_i \in \mathbb{Z}_m^t$ be the indicator vector of the set, i.e., $u'_i = (a_1, a_2, \ldots a_t)$, where $a_i = 1$ if $i \in A$ and $a_i = 0$ otherwise. In order to simplify the construction let us add an additional coordinate which is always one i.e., $u'_i = (a_1, a_2, \ldots a_t, 1)$. Clearly $\langle u'_i, u'_i \rangle = 0$ since u'_i has exactly m ones and $\langle u'_i, u'_j \rangle = 1 + |A_i \cap A_j| \neq 0$. Since intersection of two different subsets of size $m-1$ is always less than $m-1$.

Now we want to change these vectors such that the inner product of two such vectors will be in some small set S. By the chinese reminder theorem $\mathbb{Z}_m \approx \mathbb{Z}_{p_1} \oplus \mathbb{Z}_{p_2} \ldots \oplus \mathbb{Z}_{p_r}$. Thus any number x in $\mathbb{Z}_m$ we can view as $(x \mod p_1, x \mod p_2, \ldots, x \mod p_r)$. The set S is the set $\{0,1\}^r \backslash (0, 0, \ldots 0)$ i.e. $a \in S$ iff $a \neq 0$ and for every $k = 1, \ldots r$ holds $(a \mod p_k) \in \{0, 1\}$.

By the chinese reminder theorem there exist constants $c_1, c_2 \ldots c_r \in \mathbb{Z}_m$ such that:

1. $c_i \equiv 1 \mod p_i$
2. $c_i \equiv 0 \mod p_j$ for $i \neq j$

Let us define u_i by:

$$u_i = (c_1 u_i'^{\otimes p_1 - 1}, c_2 u_i'^{\otimes p_2 - 1}, \ldots, c_r u_r'^{\otimes p_r - 1}).$$

Now we need to prove that: $\langle u_i, u_i \rangle \equiv 0$:

$$\langle u_i, u_i \rangle = \langle (c_1 u_i'^{\otimes p_1 - 1}, c_2 u_i'^{\otimes p_2 - 1}, \ldots c_r u_r'^{\otimes p_r - 1}), (c_1 u_i'^{\otimes p_1 - 1}, c_2 u_i'^{\otimes p_2 - 1}, \ldots c_r u_r'^{\otimes p_r - 1}) \rangle = \\ \sum_{j=1}^{r} c_j^2 \langle u_i'^{\otimes p_j - 1}, u_i'^{\otimes p_j - 1} \rangle = \sum_{j=1}^{r} c_j^2 \langle u_i', u_i' \rangle^{p_j - 1},$$

where the last equation follows from Fact 2.6. Since $\langle u_i', u_i' \rangle = 0$ it follows that $\langle u_i, u_i \rangle = 0$. Now let us prove that $\langle u_i, u_j \rangle \in S$ for any $i \neq j$. In order to prove that $\langle u_i, u_j \rangle \in S$ we will prove that $\langle u_i, u_j \rangle \mod p_k \in \{0, 1\}$ and $\langle u_i, u_j \rangle \neq 0$. Observe that

$$u_i \mod p_k \equiv (0, 0, \ldots, u_i'^{\otimes (p_k - 1)}, 0, \ldots, 0).$$

Thus it follows that:

$$\langle u_i, u_j \rangle \mod p_k \equiv \langle u_i'^{\otimes p_k - 1}, u_j'^{\otimes p_k - 1} \rangle \equiv \langle u_i', u_j' \rangle^{p_k - 1}$$

By Fermat's Little Theorem $x^{p_k - 1} \equiv 0$ or $1 \mod p_k$ for every k. Since $\langle u_i', u_j' \rangle \neq 0 \mod m$ for some k $\langle u_i', u_j' \rangle \neq 0 \mod p_k$. Therefore $\langle u_i, u_j \rangle = \langle u_i', u_j' \rangle^{p_k - 1} \neq 0 \mod p_k$. Therefore $\langle u_i, u_j \rangle \neq 0 \mod m$. □

As a corollary we get:

Corollary A.2. *For every h, r there exists integer $m = p_1 p_2 \ldots p_r$ and a set $S \subset \mathbb{Z}_m$ of size $2^r - 1$ and a family of S-matching vectors $\{u_i\}_{i=1}^n, u_i \in (\mathbb{Z}_m)^h$ such that $n \geq \exp(c \frac{(\log h)^r}{\log \log^{r-1} h})$.*

Note that the only difference between Corollary A.2 and Corollary 3.3 is in order of quantifiers i.e. Corollary 3.3 holds for every m while Corollary A.2 holds for some specific m.

Proof of Corollary A.2. Let us take all primes of the same size (i.e. $p_i = p_j + o(p_i)$) and $t = m^2$ then in Lemma A.1 we will get that $n \geq \binom{m^2}{m-1} \geq m^m = O(m^{p^r})$ and $h = O(m^{2p_r})$. Thus it follows that:

$$n \geq \exp(c \frac{(\log h)^r}{\log \log^{r-1} h}).$$

□