Electronic Colloquium on Computational Complexity, Report No. 69 (2008)

3-Query Locally Decodable Codes of Subexponential Length

Klim Efremenko *

August 5, 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 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 infinity many Mersenne primes. In this paper we give an unconditional 3-query LDC construction with 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 [Gro00] which holds 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 two queries LDC tight lower bounds of $2^{\theta(n)}$ were given for linear codes in [GKST02] and for arbitrary codes in [KdW03]. The Katz and Trevisan [KT00] establish lower bounds of $\tilde{\Omega}(n^2)$ for LDC with 3 queries and $\tilde{\Omega}(n^{1+1/(\lceil q/2\rceil-1)})$ for any number of queries q, and [Woo07] gives a slightly improves this bound.

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].

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

ISSN 1433-8092

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 known Mersenne primes, Yekhanin also obtains unconditional results significantly improving previous results on LDC's(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.

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 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 follow this approach. The main ingredient in our construction is the existence of super-polynomial size set-systems with restricted intersections [Gro00], which holds only over composite numbers. The codes we have constructed have a perfectly smooth decoder and thus they immediately imply 3-server Private Information Retrieval with communication complexity $\exp(O(\sqrt{\log n \log\log n}))$. This is a preliminary version. We hope to post a revision in a couple of months.

2 Definitions and Basic Facts

We will use the following standard mathematical notation:

- $[s] = \{1, \ldots s\}$;
- $\mathbb{F}_q = GF(q)$ is a finite field of q elements;
- $\mathbb{F}^*$ is a multiplicative group of the field.
- $\mathbb{Z}_N = \mathbb{Z}/N\mathbb{Z}$;
- $d_H(x, y)$ 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 $i = 1, 2, \ldots n$ such that for all $i = 1, 2, \ldots n$ the following holds

1. For every message $\vec{x} = (x_1, x_2, \ldots x_n) \in \mathbb{F}^n$ 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 $i = 1, 2, \ldots m-1$.*

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 $2^t - 1$ 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. □

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. The second part is a construction of an S-decoding polynomial with a small number of monomials, which correspond to "algebraically nice" sets. Let us fix some composite number m for our construction. We give 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 $\{u_i\}_{i=1}^n$ is said to be *S-matching* if:

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

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 $m = p_1 p_2 \ldots p_r$ 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 such that*

- $|\mathcal{H}| \geq \exp(c\frac{(\log h)^r}{\log \log^{r-1} h})$,
- *Size of every set H in set-system $\mathcal{H}$: divisible by m i.e. $|H| \equiv 0 \mod m$,*

- *Size of intersection of every two sets G, H is set-system $\mathcal{H}$ modulo m is in set S. i.e.* $\forall G, H \in \mathcal{H}\ G \neq H\ \ |G \cap H| \in S \mod m$, *where S is a set of size $2^r - 1$ and $0 \notin S$.*
- $\forall s \in S$ *for all* $i = 1, 2, \ldots r\ \ s \mod p_i$ *is* 0 *or* 1.

From this lemma it easily follows that:

Corollary 3.3. For every h, 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. For each set $H \in \mathcal{H}$ we will have one vector u_i which is the indicator vector of H. □

Note that in this construction, in contrast to [Yek08], we can take m to be constant and $h \to \infty$. For 3-query LDC's we will be mainly interested in the case $r = 2$ and S of size 3.

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. For constructing a 3-query LDC we will set $m = 511 = 2^9 - 1 = 7 \cdot 73$, $\mathbb{F} = GF(2^9)$ and γ is any generator of the multiplicative group $\mathbb{F}^*$. We will first construct a linear code over the field $\mathbb{F}$. We will show in the next section 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

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

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 and the general definition, by the linearity of C, is $C(\sum c_i e_i) \triangleq \sum c_i C(e_i)$. The encoding of e_i is

$$C(e_i) \triangleq (\gamma^{<u_i, x>})_{x \in (\mathbb{Z}_m)^h}. \tag{1}$$

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

$$C((c_1, c_2, \ldots c_n)) \triangleq \sum_{i=1}^{n} c_i f_i, \tag{2}$$

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

We 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 $P(x) = a_0 + a_1 x^{b_1} + a_2 x^{b_2} \ldots a_{k-1} x^{b_{k-1}}$.
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 $w(v), w(v + b_1 u_i), \ldots w(v + b_{k-1} u_i)$.
- Output

$$c_i = \gamma^{-<u_i,v>} \left(a_0 w(v) + a_1 w(v + b_1 u_i) \ldots a_{k-1} w(v + b_{k-1} u_i)\right). \tag{3}$$

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 $v, v+b_1 u_i, \ldots v+b_{k-1} u_i$ 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 $j \neq i$.

$$d_i(C(e_i)) = (\gamma^{-<u_i,v>})(a_0 \gamma^{<u_i,v>} + a_1 \gamma^{<u_i,v+b_1 u_i>} + \ldots + a_{k-1} \gamma^{<u_i,v+b_{k-1} u_i>}).$$

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

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

Now let us prove that

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

We need to show that

$$a_0 \gamma^{<u_i,v>} + a_1 \gamma^{<u_i,v+b_1 u_j>} + \ldots + a_{k-1} \gamma^{<u_i,v+b_{k-1} u_j>} = 0.$$

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

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

□

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 $(k, \delta, k\delta)$ locally decodable code $C : F^n \mapsto F^N$ with rate $\exp(\exp(O(\sqrt[r]{\log n \log\log^{r-1} n})))$ and $k \leq 2^r$.*

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

Let us give a concrete example which will allow us make 3-query LDC. We found it 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 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 $a \in \mathbb{F}_{2^t}$ exists an n by n matrix M_a over $\mathbb{F}_2$ such that $M_a x = ax$.

Assume now that we have message $(c_1, c_2, \dots, c_n) \in \mathbb{F}_2^n$ first we will view it as a message in $(\mathbb{F}_{2^t})^n$. Now let $w = C(c_1, c_2, \dots c_n)$, $w \in (\mathbb{F}_{2^t})^{m^h}$ be an encoding of the message as in the previous section. Next let us extend our codeword to be a concatenation of k identical codewords $w_0 \circ w_1 \circ w_{k-1} = w \circ w \circ \dots \circ w$. 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 k (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 $P(x) = a_0 + a_1 x^{b_1} + a_2 x^{b_2} \dots a_{k-1} x^{b_{k-1}}$ be an S-decoding polynomial. Next let us now set our codeword to be

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

where $w = C(x)$ and $a_i w$ is a coordinate wise scalar multiplication. Recall that from Equation 3 we can decode the i-th symbol c_i using the identity:

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

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

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

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 $i = 1 \dots n$ $\Pr_v(L(\gamma^{<u_i,v>}) = 1) \geq 1/2$.

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

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

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 $(c_1, c_2, \dots c_n)$ encode it , by code from previous section $w = C(c_1, c_2, \dots, c_n)$.

2. Extend it to
$$\tilde{w} \triangleq \tilde{w_0} \circ \tilde{w_1} \circ \ldots \circ \tilde{w}_{k-1} \triangleq a_0 w \circ a_1 w \ldots \circ a_{k-1} w.$$

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

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 $w_0(v), w_1(v + b_1 u_i), \ldots, w_{k-1}(v + b_{k-1} u_i)$.
- Output $c_i = w_0(v) \oplus w_1(v + b_1 u_i) \ldots \oplus w_{k-1}(v + b_{k-1} u_i)$.

Algorithm 2: Decoding Algorithm

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

Proof. We will prove it in two steps.

First let us prove that if at most δ fraction of the codeword $w = w_0 \circ w_1 \ldots \circ w_{k-1}$ is damaged then we query a damaged place with probability at most $2k\delta$. Let δ_i be a fraction of damaged bits in w_i so $\frac{1}{k}\sum \delta_i = \delta$. 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 $\sum 2\delta_i = 2k\delta$.

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 $i \neq j$.

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

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 $c_i = 0$.

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

□

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.

6 ACKNOWLEDGMENTS

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 Oded Goldreich, Ariel Gabizon, Omer Reingold, Shachar Lovett, and my wife Rivka for their valuable comments.

References

[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.

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

[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.

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

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

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

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

ECCC ISSN 1433-8092

http://eccc.hpi-web.de/