Electronic Colloquium on Computational Complexity, Report No. 20 (2015)

Explicit Strong LTCs with inverse poly-log rate and constant soundness

Michael Viderman[*]

January 31, 2015

Abstract

An error-correcting code $C \subseteq \mathbb{F}^n$ is called (q, ϵ)-strong locally testable code (LTC) if there exists a tester that makes at most q queries to the input word. This tester accepts all codewords with probability 1 and rejects all non-codewords $x \notin C$ with probability at least $\epsilon \cdot \delta(x, C)$, where $\delta(x, C)$ denotes the relative Hamming distance between the word x and the code C. The parameter q is called the query complexity and the parameter ϵ is called soundness.

Goldreich and Sudan (J.ACM 2006) asked about the existence of strong LTCs with constant query complexity, constant relative distance, constant soundness and inverse polylogarithmic rate. They also asked about the explicit constructions of these codes.

Strong LTCs with the required range of parameters were obtained recently in the works of Viderman (CCC 2013, FOCS 2013) based on the papers of Meir (SICOMP 2009) and Dinur (J.ACM 2007). However, the construction of these codes was *probabilistic*.

In this work we show that codes presented in the works of Dinur (J.ACM 2007) and Ben-Sasson and Sudan (SICOMP 2005) provide the *explicit* construction of strong LTCs with the above range of parameters. Previously, such codes were proven to be weak LTCs. Using the results of Viderman (CCC 2013, FOCS 2013) we prove that such codes are, in fact, strong LTCs.

[*]System Validation Engineering, Intel Israel, Haifa, 31015, Israel. Email: michael.viderman@intel.com

ISSN 1433-8092

Contents

1 Introduction

Probabilistically Checkable Proof (PCP) systems [2, 3, 20] (a.k.a. Holographic Proofs [4]) are proof systems that allow efficient probabilistic verification of a claim by reading few symbols of the proof. The celebrated PCP theorem [2, 3] is one of the main breakthrough results in complexity theory. This theorem asserts that for every language in $\mathcal{NP}$ there exists a polynomial-time PCP verifier that queries the proof in a constant number of locations. The verifier is guaranteed to always accept valid proofs of true statements, and to accept any claimed proof of false assertions with low probability. The theorem has found many applications in theoretical computer science, especially in establishing lower bounds for approximation algorithms [6, 5, 20, 24].

Informally, most of the PCP constructions were achieved using error-correcting codes, possessing nice properties. Let us first give some auxiliary definitions regarding error-correcting codes.

A code over a finite alphabet Σ is a subspace $\mathcal{C} \subseteq \Sigma^n$. A linear code over a finite field $\mathbb{F}$ is a linear subspace $\mathcal{C} \subseteq \mathbb{F}^n$. In this case, n is the blocklength of the code $\mathcal{C}$, denoted by $\text{blocklength}(\mathcal{C})$. The dimension of a linear code $\mathcal{C}$, denoted by $\dim(\mathcal{C})$, is its dimension as a vector space and is equal to $\log_{|\mathbb{F}|} |\mathcal{C}|$. The dimension of a non-linear code $\mathcal{C}$ over the alphabet Σ is defined to be $\dim(\mathcal{C}) = \log_{|\Sigma|} |\mathcal{C}|$. The rate of a code $\mathcal{C}$, denoted by $\text{rate}(\mathcal{C})$, is defined to be $\frac{\dim(\mathcal{C})}{\text{blocklength}(\mathcal{C})} = \frac{\dim(\mathcal{C})}{n}$.

We define the distance between two words $x, y \in \mathbb{F}^n$ to be $\Delta(x,y) = |\{i \mid x_i \neq y_i\}|$ and the relative distance to be $\delta(x,y) = \frac{\Delta(x,y)}{n}$. The distance of $\mathcal{C}$ is defined by $\Delta(\mathcal{C}) = \min_{x \neq y \in \mathcal{C}} \Delta(x,y)$ and its relative distance is defined by $\delta(\mathcal{C}) = \frac{\Delta(\mathcal{C})}{n}$. We note that if $\mathcal{C}$ is linear then $\Delta(\mathcal{C}) = \min_{c \in \mathcal{C} \setminus \{0\}} \{|c|\}$. One is typically interested in codes whose distance is linear to the blocklength of $\mathcal{C}$, i.e., $\Omega(n)$.

For $x \in \mathbb{F}^n$ and $\mathcal{C} \subseteq \mathbb{F}^n$, let $\delta(x, \mathcal{C}) = \min_{y \in \mathcal{C}} \{\delta(x,y)\}$ denote the relative distance of x from the code $\mathcal{C}$. If $\delta(x, \mathcal{C}) \geq \rho$, we say that x is ρ-far from $\mathcal{C}$ and otherwise x is ρ-close to $\mathcal{C}$.

1.1 Locally Testable Codes

Most of the PCP constructions (e.g., [7, 11, 17, 22]) are tightly related to a special kind of error-correcting codes possessing some testability properties. These codes are called *locally testable*.

In other words, locally testable codes (LTCs) are error correcting codes that have a tester, which is a randomized algorithm with oracle access to the received word x. The tester reads a sublinear amount of information from x and based on this "local view" decides if $x \in C$ or not. It should accept codewords with probability one, and reject words that are far (in Hamming distance) from the code with noticeable probability. Such codes are of interest in computer science due to their numerous connections to probabilistically checkable proofs (PCPs) and property testing (see the surveys [34, 21] for more information). LTCs were implicit already in [4] (cf. [21, Sec. 2.4]) and they were explicitly studied by Goldreich and Sudan [22].

By now several different constructions of LTCs are known including codes based on low-degree polynomials over finite fields and affine-invariant codes [1, 2, 16, 9, 8, 15, 23, 26, 28, 25, 32], constructions based on PCPs of proximity/assignment testers [7, 18, 17][1], sparse random linear codes [14, 27, 29] and tensor products of codes [19, 13, 12, 30, 35].

Basically, there are two kinds of LTCs: weak and strong. A code $\mathcal{C}$ is said to be (q, ϵ, ρ)-weak LTC if there exists a randomized algorithm T, called tester, that makes at most q queries to the

[1] As was pointed out in [22], not all PCP constructions are known to yield LTCs, but some of them (e.g., PCPs of proximity/assignment testers) can be adapted to yield LTCs.

input word w. If $w \in \mathcal{C}$ then T accepts w with probability 1, but if w is ρ-far from C the tester T rejects w with probability at least ϵ. Let us notice that the tester is not required to reject when $0 < \delta(w, \mathcal{C}) < \rho$. This is the reason why such codes are called *weak* LTCs.

In contrast to weak LTCs, the testers for strong LTCs are required to reject all non-codewords with corresponding probability. More formally, a code $\mathcal{C}$ is called (q, ϵ)-strong LTC if there exists a tester T that makes at most q queries to the input word w. If $w \in \mathcal{C}$ then T accepts w with probability 1, but if $w \notin \mathcal{C}$ then T rejects w with probability at least $\epsilon \cdot \delta(w, \mathcal{C})$. The parameter q is called the query complexity and the parameter ϵ is called soundness.

Informally, we say that a code $\mathcal{C}$ is a weak LTC if it has a linear distance and there exist constants $q, \epsilon > 0$ and $\rho \leq \delta(\mathcal{C})/3$ such that C is a (q, ϵ, ρ)-weak LTC. [2] Similarly, we say that a code $\mathcal{C}$ is a strong LTC if it has a linear distance and there exist constants $q, \epsilon > 0$ such that $\mathcal{C}$ is a (q, ϵ)-strong LTC.

LTCs were explicitly studied in the work of Goldreich and Sudan [22], who presented probabilistic construction of strong LTCs. These LTCs achieve constant query complexity, constant soundness and rate $\dfrac{1}{\exp(\tilde{O}(\sqrt{\log n}))}$, where n denotes the blocklength.

Later, other constructions of LTCs [11, 17, 30] succeeded to obtain the rate $\dfrac{1}{\mathrm{polylog}(n)}$ together with constant query complexity and soundness, however these codes were weak LTCs. It can be verified that every strong LTC is also a weak LTC, but some weak LTCs are not strong LTCs [36]. In the journal version of [22], the authors pointed out that all known LTCs that achieve inverse polylogarithmic rate are weak LTCs, and asked about the existence of strong LTCs with polylogarithmic rate and, in particular, about the *explicit* construction of such codes [22, Section 6].

The previous papers of the author [36, 37] showed a *probabilistic* construction of binary linear 3-query strong LTCs with inverse polylogarithmic rate, constant soundness and constant relative distance. In this paper (Section 1.3), we show the explicit construction of linear strong LTCs with constant query complexity, constant soundness, polylogarithmic rate and constant relative distance over a fixed field, therefore resolving a question raised by Goldreich and Sudan [22].[3]

1.2 Preliminaries

Let $[n]$ be the set $\{1, \ldots, n\}$. For $w \in \mathbb{F}^n$, let $\mathrm{supp}(w) = \{i \in [n] \mid w_i \neq 0\}$ and $|w| = |\mathrm{supp}(w)|$. For $u = (u_1, u_2, \ldots, u_n)$, $v = (v_1, v_2, \ldots, v_n) \in \mathbb{F}^n$ let $\langle u, v \rangle$ denote the bilinear function from $\mathbb{F}^n \times \mathbb{F}^n$ to $\mathbb{F}$ defined by $\langle u, v \rangle = \sum_{i=1}^{n} u_i v_i$. The dual code is defined by $\mathcal{C}^\perp = \{u \in \mathbb{F}^n \mid \forall c \in \mathcal{C} : \langle u, c \rangle = 0\}$. Similarly, we define $\mathcal{C}^\perp_{\leq t} = \left\{u \in \mathcal{C}^\perp \mid |u| \leq t\right\}$. For $w \in \mathbb{F}^n$ and $S = \{j_1, j_2, \ldots, j_m\} \subseteq [n]$ we let $w|_S = (w_{j_1}, w_{j_2}, \ldots, w_{j_m})$, where $j_1 < j_2 < \ldots < j_m$, be the restriction of w to the subset S. Similarly, we let $\mathcal{C}|_S = \{c|_S \mid c \in \mathcal{C}\}$ denote the projection of the code $\mathcal{C}$ onto S. We define

[2] The parameter ρ is required to be less than $\delta(\mathcal{C})/2$ to avoid trivial solutions like claiming that every perfect code $\mathcal{C}$ is a $(0, 1, \delta(\mathcal{C})/2)$-weak LTC. Recall that a code $\mathcal{C} \subseteq \mathbb{F}^n$ is called perfect if there are no words in $\mathbb{F}^n$ that are $(\delta(\mathcal{C})/2)$-far from $\mathcal{C}$. So, in this case one could say that no queries are needed and all $(\delta(\mathcal{C})/2)$-far words are rejected with probability 1 vacuously.

[3] A suggestion to show such explicit construction was raised in personal discussion with Or Meir, and later was asked in [37].

$\mathcal{C}|_{-S} = \mathcal{C}|_{[n]\setminus S}$, i.e., projection of the code $\mathcal{C}$ to all coordinates besides S. For $A \subseteq \mathbb{N}$ and $b \in \mathbb{N}$ we let $A + b = b + A = \{a + b \mid a \in A\}$.

For the distribution $\mathcal{D}$ over the subsets of $[n]$ we let $\mathcal{D}(I)$ to denote the probability that a subset $I \subseteq [n]$ is selected by $\mathcal{D}$ and $\text{supp}(\mathcal{D}) = \{I \subseteq [n] \mid \mathcal{D}(I) > 0\}$. For $i \in [n]$ we let $N_{\mathcal{D}}(i) = \{I \in \text{supp}(\mathcal{D}) \mid i \in I\}$.

Now we define testers and LTCs (see [22, 36] for the justification of this definition).

Definition 1.1 (LTCs and Testers)**.** A q-query tester for a code $\mathcal{C} \subseteq \mathbb{F}^n$ is a distribution $\mathcal{D}$ over subsets $I \subseteq [n]$ such that $|I| \leq q$. A q-query tester $\mathcal{D}$ is a (q, ϵ, ρ)-weak tester if for all $w \in \mathbb{F}^n$, $\delta(w, \mathcal{C}) \geq \rho$ we have $\underset{I \sim \mathcal{D}}{\mathbf{Pr}}[w|_I \notin \mathcal{C}|_I] \geq \epsilon$. A q-query tester $\mathcal{D}$ is a (q, ϵ)-strong tester if for all $w \in \mathbb{F}^n$ we have $\underset{I \sim \mathcal{D}}{\mathbf{Pr}}[w|_I \notin \mathcal{C}|_I] \geq \epsilon \cdot \delta(w, \mathcal{C})$.

A code $\mathcal{C} \subseteq \mathbb{F}^n$ is a (q, ϵ, ρ)-*weak LTC* if it has a (q, ϵ, ρ)-weak tester. A code $\mathcal{C} \subseteq \mathbb{F}^n$ is a (q, ϵ)-*strong LTC* if it has a (q, ϵ)-strong tester.

Remark 1.2. Although the tester in Definition 1.1 does not output accept or reject, the way a standard tester does, it can be converted to output accept, reject as follows. Whenever the task is to test whether $w \in \mathcal{C}$ and a subset $I \subseteq [n]$ is selected by the tester, the tester can output accept if $w|_I \in \mathcal{C}|_I$ and otherwise output reject. In this manner, the tester always accepts the codewords of $\mathcal{C}$.

It is not hard to see that every strong LTC is a weak LTC, but not vice versa [37].

1.3 Main Results

In this paper we show the *explicit* construction of strong LTCs with a range of parameters asked by Goldreich and Sudan [22]. Although the requested range of parameters was achieved for the *probabilistic* construction of strong LTCs [36, 37], explicit strong LTCs with this range of parameters was not obtained.

Theorem 1.3 (Main Theorem)**.** *There exist constants $q, d, \epsilon, \gamma > 0$ and a constant size field $\mathbb{F}$ such that for infinitely many $n \in \mathbb{N}^+$ we have an* explicit *construction of a linear code $C \subseteq \mathbb{F}^n$, where*

- *C is a (q, ϵ)-strong LTC,*
- *$\delta(C) \geq \gamma$ and* $\text{rate}(C) \geq \frac{1}{\log^d n}$.

The proof of Theorem 1.3 is given in Section 2.

2 Proof of Theorem 1.3

The proof of Theorem 1.3 contains three parts. In Section 2.1 we recall the notion of relaxed LTCs and some related results [37]. In particular, this section shows that in order to get desired explicit string LTCs it is sufficient to get explicit relaxed LTCs with sufficiently good parameters range (see Corollary 2.5).

A second part is presented in Section 2.2, where it is explained that the result of Ben-Sasson and Sudan [11] gives explicit relaxed LTCs, however their parameters range is not sufficiently nice.

Finally, Section 2.3 shows that a work of Dinur [17] can be used to improve that soundness of the relaxed LTCs. It turns out that this improvement is sufficient to change the codes of [11] to relaxed LTCs with sufficiently good parameters (which is what we need by Corollary 2.5). Since the gap amplification technique is explicit, it yields explicit construction of relaxed LTCs.

2.1 Relaxed LTCs.

First, we recall a notion of *relaxed LTCs* [37]. Intuitively, relaxed LTCs have two kind of coordinates: those with good testability and those which worse (but non-trivial) testability (see Definition 2.2). Then, we recall an observation and its corollary (Corollary 2.5) of [37] saying that such relaxed LTCs can be easily converted to strong LTCs. Hence, all we need to prove Theorem 1.3 is to construct relaxed LTCs with a corresponding range of parameters.

Before we present Observation 2.4, we recall some concept used in [37].

Definition 2.1 (A core of the code)**.** Let $C \subseteq \Sigma^n$ be a code. A core of the code C, denoted by $A(C)$, is a nonempty subset of $[n]$ such that if $A(C) \neq [n]$ then any assignment to the entries of $A(C)$ uniquely determines the entries of $[n] \setminus A(C)$ and vice versa. I.e., if $A(C) \neq [n]$ then for any $c \in C$ there is no $c' \in C$ such that $c|_{A(C)} = c'|_{A(C)}$ and $c|_{[n]\setminus A(C)} \neq c'|_{[n]\setminus A(C)}$, or $c|_{[n]\setminus A(C)} = c'|_{[n]\setminus A(C)}$ and $c|_{A(C)} \neq c'|_{A(C)}$.

Clearly, there might be many options for $A(C)$, and in this case we fix only one such option. If $A(C) = [n]$ then for any $w, w' \in \Sigma^n$ we let $\delta(w|_{[n]\setminus A(C)}, w'|_{[n]\setminus A(C)}) = \delta(w|_{[n]\setminus A(C)}, C|_{[n]\setminus A(C)}) = 0$.

Now we recall the concept of a relaxed LTC (rLTC).

Definition 2.2 (Relaxed LTC)**.** A q-query tester $\mathcal{D}$ is a $(q, \epsilon_1, \epsilon_2)$-rLTC tester for a linear code $C \subseteq \mathbb{F}^n$ with a core $A(C)$, if for every $w \in \mathbb{F}^n$ there exists $c \in C$ such that $\underset{I\sim\mathcal{D}}{\mathbf{Pr}}[w|_I \notin C|_I] \geq \max\left\{\epsilon_1 \cdot \delta(w|_{A(C)}, c|_{A(C)}), \epsilon_2 \cdot \delta(w|_{-A(C)}, c|_{-A(C)})\right\}$. A code $C \subseteq \mathbb{F}^n$ with a core $A(C)$ is a $(q, \epsilon_1, \epsilon_2)$-*rLTC* if it has a $(q, \epsilon_1, \epsilon_2)$-rLTC tester.

The parameter q is called the query complexity, ϵ_1 is called the first soundness parameter and ϵ_2 is called the second soundness parameter.

Intuitively, think that ϵ_1 is a constant, but ϵ_2 is sub-constant.

The following simple observation says that any strong LTC is also a relaxed LTC with similar parameters.

Observation 2.3 (Strong LTCs are relaxed)**.** *If $C \subseteq \mathbb{F}^n$ is a (q, ϵ)-strong LTC then it is also a $(q, \epsilon, 1)$-rLTC with regards to the code $A(C) = [n]$.*

The observation follows immediately from the definition of relaxed LTCs (Definition 2.2).

An observation made in [37] was that a relaxed LTC with sub-constant second soundness parameter can be easily converted to a strong LTC with a constant soundness.

Observation 2.4 (A conversion of rLTCs to strong LTCs)**.** *Let $q \geq 2$ and $C \subseteq \mathbb{F}^n$ be a linear $(q, \epsilon_1, \epsilon_2)$-rLTC with a core $A(C)$. Then there exists a linear $(q, \epsilon_1/6)$-strong LTC $C' \subseteq \mathbb{F}^{n'}$, where $n \leq n' \leq \frac{12}{\epsilon_2} \cdot n$, $\dim(C') = \dim(C)$, $\mathrm{rate}(C') \geq \frac{\epsilon_2}{12} \cdot \mathrm{rate}(C)$ and $\delta(C') \geq 0.9 \cdot \delta(C|_{A(C)})$. Moreover, the construction of C' from C is explicit and done in time $O(n')$.*

Based on Observation 2.4, [37] proved the following corollary that will play a crucial role in the proof of Theorem 1.3.

Corollary 2.5. *Assume that for constants $q \geq 2, \epsilon > 0$, field $\mathbb{F}$ and infinitely many $n \in \mathbb{N}^+$ we have a linear code $C \subseteq \mathbb{F}^n$ with a core $A(C)$ such that C is a $(q, \epsilon, \frac{1}{\mathrm{polylog}(n)})$-rLTC, $\delta(C|_{A(C)}) = \Omega(1)$ and $\mathrm{rate}(C) = \frac{1}{\mathrm{polylog}(n)}$. Then, there exists $C' \subseteq \mathbb{F}^{n'}$ such that $n \leq n' \leq n \cdot \mathrm{polylog}(n)$, C' is a $(q, \epsilon/6)$-strong LTC, $\delta(C') = \Omega(1)$ and $\mathrm{rate}(C') = \frac{1}{\mathrm{polylog}(n')}$. Moreover, C' is constructed explicitly from C.*

2.2 Construction of [11] gives relaxed LTCs

We take the construction of Ben-Sasson and Sudan as [11] as a main ingredient of the proof. We show that such codes have stronger testability properties that were proven in [11].

To present more detailed explanation, let us define the Reed-Solomon codes.

Definition 2.6 (Reed-Solomon codes)**.** Let K denote a finite field, let $S \subseteq K$ and let $d < |S|$ denote a natural number. The Reed-Solomon code $RS_{K,S,d} : K^{d+1} \to K^{|S|}$ is defined as follows: Suppose we wish to encode a message $a \in K^{d+1}$ with $RS_{K,S,d}$. We define the polynomial $P_a(X) = \sum_{i=0}^{d} a_i X^i$, and set the codeword $RS_{K,S,d}(a)$ to consist of the evaluations of P_a at each of the elements of S. The relative distance of $RS_{K,S,d}$ is $1 - \frac{d+1}{|S|}$ (see Lecture 4 in [33]).

In [11] it was shown that certain Reed-Solomon codes can provide weak LTCs (using repetitions of some coordinates). More precisely, they showed the following result (the statement uses a concept CWP, defined in [30], which is almost identical to the PCP of proximity).

Theorem 2.7 (Theorem 4 in [11])**.** *Let $K = GF(2^l)$ and let $L \subseteq K$ be a $GF(2)$-linear subspace of K. Then for any $d < |L|$ the code $RS_{K,L,d}$ is a CWP with query complexity $O(1)$, rejection ratio $1/\mathrm{poly}(\log |L|)$, randomness complexity $\log |L| + O(\log \log |L|)$ and proof length $|L| \cdot \mathrm{poly}(\log |L|)$.*

We note that CWP gives immediately a weak LTC with similar parameters range by repetition of code coordinates a number of times [30].

The parameters range we are interested in is obtained by choosing $d = O(|L|)$, so one gets a CWP with constant query complexity and inverse poly-log rejection ratio, with an alphabet of a super-constant size (since a Reed-Solomon code of block length n must be over an alphabet of size at least n).

As was pointed out by Meir [30], the construction of Ben-Sasson and Sudan [11] can be viewed as iterative construction of Reed-Solomon codes, where every step, a tensor product and a deterministic projection are applied. Besides being explicit, such a construction has similar properties to the construction of [36] and it can be proved similarly, that it gives strong LTCs with inverse poly-log soundness. More formally, the following theorem can be proved.

Theorem 2.8. *For some constant $q, d \in \mathbb{N}^+$, a fixed field $\mathbb{F}$ and infinitely many $n \in \mathbb{N}^+$ there exists an explicit linear code $C \subseteq \mathbb{F}^n$ and its tester $\mathcal{D}$ such that*

- *C is a $(q, \frac{1}{\log^d n})$-strong LTC with respect to $\mathcal{D}$,*
- *$\delta(C) = \Omega(1)$,*
- *$\mathrm{rate}(C) = \frac{1}{\log^d n}$,*
- *$|\mathrm{supp}(\mathcal{D})| \leq n \log^d n$ and for every $u \in \mathrm{supp}(\mathcal{D})$ it holds that $\mathcal{D}(u) \leq \frac{\log^d n}{n}$, and*
- *for every $i \in [n]$ we have $|N_{\mathcal{D}}(i)| \leq \log^d n$.*

The proof of Theorem 2.8 appears in Section 3.

Although the construction of [11] is done over a large field, it can be reduced to the field of constant size, where the blocklength is increased by poly-log factor and the soundness parameter is decreased by poly-log factor. This preserves the strong LTC to have inverse poly-log rate and soundness.

We notice that by Observation 2.3, Theorem 2.8 gives also explicit $(O(1), \frac{1}{\mathrm{polylog}(n)}, 1)$-rLTCs with the same properties stated in the Theorem.

2.3 Gap Amplification of Dinur [17]

In [37] it was explained that the gap amplification of Dinur can be applied to strong LTCs with inverse poly-log soundness, which can be considered as a relaxed LTC (Observation 2.3). In this case, the gap amplification preserves it to be an rLTC where first soundness parameter is increased for a constant, while the second parameter is decreased only by a poly-log factor. In particular, Section 4 and Section 5 in [37] show that the gap amplification technique can be applied to the codes from Theorem 2.8 such that the following theorem follows.

Theorem 2.9. *For constant $q \geq 2, \epsilon > 0$, a fixed field $\mathbb{F}$ and infinitely many $n \in \mathbb{N}^+$ we have explicit construction for a linear code $C \subseteq \mathbb{F}^n$ with a core $A(C)$ such that C is a $(q, \epsilon, \frac{1}{\text{polylog}(n)})$-rLTC, $\delta(C|_{A(C)}) = \Omega(1)$ and $\text{rate}(C) = \frac{1}{\text{polylog}(n)}$.*

Theorem 1.3 follows from Theorem 2.9 and Corollary 2.5.

3 Proof of Theorem 2.8

First, let us state one of the results in the work of Polishchuk and Spielman [31].

Theorem 3.1 (Theorem 9 in [31])**.** *Let $\mathbb{F}$ be a field, let $X = \{x_l, ..., x_n\} \subseteq \mathbb{F}$, and let $Y = \{y_l, ..., y_n\} \subseteq \mathbb{F}$. Let $R(x, y)$ be a polynomial over $\mathbb{F}$ of degree (d, n) and let $C(x, y)$ be a polynomial over $\mathbb{F}$ of degree (n, d). If*

$$\Pr_{(x,y)\in X\times Y}[R(x, y) \neq C(x, y)] < \delta^2,$$

and $n > 2\delta n + 2d$, then there exists a polynomial $Q(x, y)$ of degree (d, d) such that

$$\Pr_{(x,y)\in X\times Y}\left[R(x, y) \neq Q(x, y) \quad or \quad C(x, y) \neq Q(x, y) < 2\delta^2.\right]$$

This result was one of the main technical ingredients which allowed a composition in the work of Ben-Sasson and Sudan [11]. Informally, it says that given a word, which is candidate to be tensoring of two single polynomials, if a typical "row" is close to d-degree polynomial, and a typical "column" is close to d-degree polynomial, then the candidate word is close to be the tensoring of two d-degree polynomials.

Now, let us recall some definitions from [36].

Definition 3.2 (Core oriented distance)**.** Assume $C \subseteq \mathbb{F}^n$ is a linear code and $A(C)$ is its core. We define a core oriented distance between two words $w, w' \in F^n$ to be

$$\delta_{A(C)}(w, w') = \max\left\{\delta(w, w'), \delta(w|_{A(C)}, w'_{A(C)})\right\}.$$

and a core oriented distance between the word $w \in \mathbb{F}^n$ and the code C to be

$$\delta_{A(C)}(w, C) = \min_{c\in C}\left\{\delta_{A(C)}(w, c)\right\}.$$

Definition 3.3 (Core Oriented LTC (COLTC))**.** Let $C \subseteq \mathbb{F}^n$ be a linear code and let $\mathcal{D}$ be a distribution over subsets $I \subseteq [n]$ such that $|I| \leq q$. A $\mathcal{D}$ is a (q, ϵ)-COLTC tester if, given that $A(C)$ is a core of C, for all $w \in \mathbb{F}^n$ we have

$$\Pr_{I\sim D}[w|_I \notin C|I] \geq \epsilon \cdot \delta_{A(C)}(w, C).$$

A code $C \subseteq \mathbb{F}^n$ is called a (q, ϵ)-COLTC if it has a (q, ϵ)-COLTC tester.

Let C be a linear code and $A(C)$ be its core. If C is a (q,ϵ)-COLTC (with respect to the tester D_C) then C is a (q,ϵ)-strong LTC. To see this let $w \in \mathbb{F}^n$ and note that

$$\mathop{\mathbf{Pr}}_{I \sim D_C}[w|_I \in C|_I] \geq \epsilon \cdot \delta_{A(C)}(w,c) \geq \epsilon \cdot \delta(w,C).$$

In [36] it was proved that tensoring of two core oriented LTCs stays to be a core oriented LTC, where the soundness parameter is reduced by a constant. Also, "random projection" always preserves this property. It is also the case with the work of Ben-Sasson and Sudan [11], where the tensoring is made over polynomials and the "projection" stage is explicit. The main difference is that in [11] the testability is preserved in the composition due to Theorem 3.1, while in [36] the testability is preserved due to [10, 35].

Thus, similarly to [36] it can be proved that the construction of [11] provides core oriented LTCs with constant query complexity and inverse poly-log soundness. Also, since the construction includes $O(\log\log(n))$ iterations of tensoring and "explicit projections", assuming we obtain the code $C \subseteq \mathbb{F}^n$ with a tester $\mathcal{D}$, it holds that for some constant $d \in \mathbb{N}^+$ we have

- $|\operatorname{supp}(\mathcal{D})| \leq n \log^d n$ and for every $u \in \operatorname{supp}(\mathcal{D})$ it holds that $\mathcal{D}(u) \leq \frac{\log^d n}{n}$, and
- for every $i \in [n]$ we have $|N_{\mathcal{D}}(i)| \leq \log^d n$.

This is true since every iteration the ratio of a support of a tester divided by the blocklength of a code is increased only by a fixed constant. Similar things hold with the probability that the tester takes a fixed test or a fixed coordinate is selected.

Acknowledgements

The author thanks Or Meir for raising the suggestion to obtain an explicit construction of strong LTCs by applying the arguments of [36, 37] to the codes of [11].

References

[1] Noga Alon, Tali Kaufman, Michael Krivelevich, Simon Litsyn, and Dana Ron. Testing Reed-Muller codes. *IEEE Transactions on Information Theory*, 51(11):4032–4039, 2005.

[2] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof Verification and the Hardness of Approximation Problems. *Journal of the ACM*, 45(3):501–555, May 1998.

[3] Sanjeev Arora and Shmuel Safra. Probabilistic Checking of Proofs: A New Characterization of NP. *Journal of the ACM*, 45(1):70–122, January 1998.

[4] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking Computations in Polylogarithmic Time. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing (STOC), May 5-8, 1991, New Orleans, Louisiana, USA*, pages 21–31. ACM, 1991.

[5] M. Bellare, S. Goldwasser, C. Lund, and A. Russell. Efficient probabilistically checkable proofs and applications to approximation. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing (STOC)*, pages 294–304, New York, 1993. ACM SIGACT, ACM Press.

[6] Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free Bits, PCPs, and Nonapproximability—Towards Tight Results. *SIAM Journal on Computing*, 27(3):804–915, June 1998.

[7] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust PCPs of Proximity, Shorter PCPs, and Applications to Coding. *SIAM Journal on Computing*, 36(4):889–974, 2006.

[8] Eli Ben-Sasson, Elena Grigorescu, Ghid Maatouk, Amir Shpilka, and Madhu Sudan. On sums of locally testable affine invariant properties. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, volume 6845 of *Lecture Notes in Computer Science*, pages 400–411. Springer, 2011.

[9] Eli Ben-Sasson, Noga Ron-Zewi, and Madhu Sudan. Sparse affine-invariant linear codes are locally testable. In *FOCS*, pages 561–570. IEEE Computer Society, 2012.

[10] Eli Ben-Sasson and Madhu Sudan. Robust locally testable codes and products of codes. *Random Struct. Algorithms*, 28(4):387–402, 2006.

[11] Eli Ben-Sasson and Madhu Sudan. Short PCPs with polylog query complexity. *SIAM J. Comput*, 38(2):551–607, 2008.

[12] Eli Ben-Sasson and Michael Viderman. Composition of Semi-LTCs by Two-Wise Tensor Products. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, volume 5687 of *Lecture Notes in Computer Science*, pages 378–391. Springer, 2009.

[13] Eli Ben-Sasson and Michael Viderman. Tensor Products of Weakly Smooth Codes are Robust. *Theory of Computing*, 5(1):239–255, 2009.

[14] Eli Ben-Sasson and Michael Viderman. Low rate is insufficient for local testability. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, volume 6302 of *Lecture Notes in Computer Science*, pages 420–433. Springer, 2010.

[15] Arnab Bhattacharyya, Swastik Kopparty, Grant Schoenebeck, Madhu Sudan, and David Zuckerman. Optimal testing of reed-muller codes. In *FOCS*, pages 488–497. IEEE Computer Society, 2010.

[16] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47(3):549–595, December 1993.

[17] Irit Dinur. The PCP theorem by gap amplification. *Journal of the ACM*, 54(3):12:1–12:44, June 2007.

[18] Irit Dinur and Omer Reingold. Assignment Testers: Towards a Combinatorial Proof of the PCP Theorem. *SIAM Journal on Computing*, 36(4):975–1024, 2006.

[19] Irit Dinur, Madhu Sudan, and Avi Wigderson. Robust Local Testability of Tensor Products of LDPC Codes. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, volume 4110 of *Lecture Notes in Computer Science*, pages 304–315. Springer, 2006.

[20] Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM*, 43(2):268–292, 1996.

[21] Oded Goldreich. Short Locally Testable Codes and Proofs (Survey). *Electronic Colloquium on Computational Complexity (ECCC)*, (014), 2005.

[22] Oded Goldreich and Madhu Sudan. Locally testable codes and PCPs of almost-linear length. *Journal of the ACM*, 53(4):558–655, July 2006.

[23] Elena Grigorescu, Tali Kaufman, and Madhu Sudan. Succinct representation of codes with applications to testing. *SIAM J. Discrete Math*, 26(4):1618–1634, 2012.

[24] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.

[25] Tali Kaufman and Shachar Lovett. New Extension of the Weil Bound for Character Sums with Applications to Coding. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, (FOCS)*, pages 788–796. IEEE, 2011.

[26] Tali Kaufman and Dana Ron. Testing polynomials over general fields. *SIAM J. Comput*, 36(3):779–802, 2006.

[27] Tali Kaufman and Madhu Sudan. Sparse Random Linear Codes are Locally Decodable and Testable. In *FOCS*, pages 590–600. IEEE Computer Society, 2007.

[28] Tali Kaufman and Madhu Sudan. Algebraic property testing: the role of invariance. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC), Victoria, British Columbia, Canada, May 17-20, 2008*, pages 403–412. ACM, 2008.

[29] Swastik Kopparty and Shubhangi Saraf. Local list-decoding and testing of random linear codes from high error. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC)*, pages 417–426. ACM, 2010.

[30] Or Meir. Combinatorial Construction of Locally Testable Codes. *SIAM J. Comput*, 39(2):491–544, 2009.

[31] Alexander Polishchuk and Daniel A. Spielman. Nearly-linear size holographic proofs. In *Proceedings of the 26th Annual Symposium on the Theory of Computing*, pages 194–203, New York, May 1994. ACM Press.

[32] Noga Ron-Zewi and Madhu Sudan. A new upper bound on the query complexity for testing generalized reed-muller codes. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, volume 7408 of *Lecture Notes in Computer Science*, pages 639–650. Springer, 2012.

[33] Madhu Sudan. Algorithmic introduction to coding theory, Lecture notes, 2001.

[34] Luca Trevisan. Some Applications of Coding Theory in Computational Complexity, September 23 2004.

[35] Michael Viderman. A combination of testability and decodability by tensor products. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, volume 7408 of *Lecture Notes in Computer Science*, pages 651–662. Springer, 2012.

[36] Michael Viderman. Strong LTCs with inverse polylogarithmic rate and soundness. *CCC 2013. Electronic Colloquium on Computational Complexity (ECCC)*, 19:168, 2012.

[37] Michael Viderman. Strong LTCs with inverse poly-log rate and constant soundness. *FOCS 2013. Electronic Colloquium on Computational Complexity (ECCC)*, 20:22, 2013.