2013 IEEE 54th Annual Symposium on Foundations of Computer Science

Strong LTCs with inverse poly-log rate and constant soundness

Michael Viderman
Computer Science Department
Technion — Israel Institute of Technology
Haifa, Israel
Email: viderman@cs.technion.ac.il

***Abstract*—An error-correcting code C 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 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.**

In this paper we resolve an open question raised by Goldreich and Sudan (J.ACM 2006) and construct binary linear strong LTCs with query complexity 3, constant relative distance, constant soundness and inverse polylogarithmic rate.

Our result is based on the previous paper of the author (Viderman, ECCC TR12-168), which presented binary linear strong LTCs with query complexity 3, constant relative distance, and inverse polylogarithmic soundness and rate. We show that the "gap amplification" procedure of Dinur (J.ACM 2007) can be used to amplify the soundness of these strong LTCs from inverse polylogarithmic up to a constant, while preserving the other parameters of these codes.

Furthermore, we show that under a conceivable conjecture, there exist asymptotically good strong LTCs with poly-log query complexity.

***Keywords*-error-correcting codes; locally testable codes; PCPs;**

I. Introduction

Probabilistically Checkable Proof (PCP) systems [1], [2], [3] (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 [1], [2] 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 [5], [6], [3], [7].

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 $\mathrm{blocklength}(\mathcal{C})$. 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 $\mathrm{rate}(\mathcal{C})$, is defined to be $\frac{\dim(\mathcal{C})}{\mathrm{blocklength}(\mathcal{C})} = \frac{\dim(\mathcal{C})}{n}$.

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 $\mathcal{C}$.

A. Locally Testable Codes

Most of the PCP constructions (e.g., [8], [9], [10], [11]) 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 [12], [13] for more information). LTCs were implicit already in [4] (cf. [13, Sec. 2.4]) and they were explicitly studied by Goldreich and Sudan [11].

By now several different constructions of LTCs are known including codes based on low-degree polynomials over finite fields and affine-invariant codes [14], [1], [15], [16], [17], [18], [19], [20], [21], [22], [23], constructions based on

0272-5428/13 $26.00 © 2013 IEEE
DOI 10.1109/FOCS.2013.43

PCPs of proximity/assignment testers [8], [24], [10][1], sparse random linear codes [25], [26], [27] and tensor products of codes [28], [29], [30], [31], [32].

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 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.

The best known strong LTCs are due to Goldreich and Sudan [11], who presented probabilistic construction of strong LTCs. These LTCs achieve constant query complexity, constant soundness and rate $\frac{1}{\exp(\tilde{O}(\sqrt{\log n}))}$, where n denotes the blocklength.

Later, other constructions of LTCs [9], [10], [31] succeeded to obtain the rate $\frac{1}{\text{polylog}(n)}$ 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 [33]. So, strong LTCs are strictly stronger objects than weak LTCs. In the journal version of [11], 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 [11, Section 6]. As was pointed out by Goldreich [34], strong LTCs correspond to proximity oblivious testers [35] whereas weak LTCs are even weaker than ordinary testers, i.e., the testers for weak LTCs are supposed to work only for a fixed value of the proximity parameter.

[1]As was pointed out in [11], 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.

[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.

The previous paper of the author [33] showed a probabilistic construction of binary linear 3-query strong LTCs with inverse polylogarithmic rate, inverse polylogarithmic soundness and constant relative distance. In this paper (Section I-C), we show how to amplify the soundness parameter of these codes from inverse polylogarithmic to constant, while preserving the other parameters of these codes, therefore resolving an open question raised by Goldreich and Sudan [11]. To increase the soundness parameter we apply the gap amplification technique of Dinur [10].

An interesting point is that the gap amplification was known to improve the soundness parameter of weak LTCs [10], [31], however it was not known to preserve the strong testability requirement, where all non-codewords are rejected with corresponding probability and not only words that are sufficiently far from the code. In more details, the gap amplification procedure outputs a code accompanied with a probabilistically checkable proof that could be translated to a weak LTC.

In [33] we conjectured that it should be possible to modify this procedure to preserve this stronger property. Surprisingly, it turns out that no modification is needed (besides adapting the gap amplification to preserve the linearity of the underlying codes, as was done in [31, Section 6.4]). In Section I-C we present formally our main result (Theorem I.4) and explain the ideas that lead to its proof.

1) Asymptotically good LTCs: The main open question in the area of LTCs is whether there exists a family of asymptotically good LTCs with constant query complexity and soundness, i.e., LTCs over a constant size alphabet that have constant query complexity, constant soundness parameter, constant rate and constant relative distance [11]. A possible approach to refute the existence of such codes was suggested in [36]. In fact, [36] conjectures that such codes do not exist and proves this conjecture under quite "strong" assumptions. It is worth to mention that during last years a non-trivial effort was made in studying the limitations of LTCs, and in particular: LTCs testable with 2 queries [37], [38], [39], [40] (which is a severe restriction), random low density parity check (LDPC) codes [41], cyclic codes [42], symmetric codes [43], [44], [45], [46], LTCs with small redundancy among its tests [47] and dense LTCs [48], [36]. Nevertheless, it seems that we are very far from resolving this problem.

Let us suppose that asymptotically good LTCs with constant query complexity and constant soundness do not exist. In this case, the most intriguing question would be "What are the best LTCs we could obtain?". To address this question we should decide how to compare different LTCs. Informally, in this subject and in the area of error-correcting codes in general, we always require a constant relative distance since otherwise even a tiny fraction of errors could modify one codeword into another. Hence we want a constant relative distance and do not allow to relax this requirement. Given

that we consider only LTCs with constant relative distance, we have 3 parameters that describe the "goodness" of LTCs: the query complexity, the soundness parameter and the rate.

Constant Soundness.: It is not hard to show that LTCs with sub-constant soundness parameter ϵ and query complexity q could be converted to LTCs with soundness $\frac{1}{2}$ and query complexity $q \cdot \lceil \frac{1}{\epsilon} \rceil$. Hence, for the sake of this discussion we can require constant soundness parameter and compare different LTCs only according to their query complexity and the rate.

Constant Query Complexity.: Recall that in Theorem I.4 we show that when query complexity is required to be constant, the rate can be inverse polylogarithmic. Informally, under assumption that asymptotically good LTCs with constant query complexity and soundness do not exist, this is the best achievable rate when query complexity, relative distance and soundness parameter are required to be constant.

Constant Rate.: Indeed, one of the most natural questions is what is the minimal query complexity if the rate and the soundness parameter of an LTC are required to be constant as well as the relative distance. In other words, what is the minimal query complexity required for the asymptotically good code to be testable. [3] For the current state of the art, we know that for every constant $\epsilon > 0$ there exist asymptotically good strong LTCs with query complexity n^{ϵ} and constant soundness parameter, where n is the blocklength of the code [49], [30], [32].

In Section I-C1 we show that under a conceivable conjecture there exist asymptotically good strong LTCs with query complexity $\mathrm{polylog}(n)$ and constant soundness parameter. Informally, this is the minimal query complexity we can hope for, under conjecture that asymptotically good LTCs with constant query complexity and soundness do not exist [36].

B. 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 $\mathcal{C}|_{-S} = \mathcal{C}|_{[n] \setminus S}$, 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 $\mathrm{supp}(\mathcal{D}) = \{I \subseteq [n] \mid \mathcal{D}(I) > 0\}$. For $i \in [n]$ we let $N_{\mathcal{D}}(i) = \{I \in \mathrm{supp}(\mathcal{D}) \mid i \in I\}$.

Now we define testers and LTCs (see [11], [33] for the justification of this definition).

Definition I.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 $\mathop{\mathbf{Pr}}_{I \sim \mathcal{D}}[w|_I \notin \mathcal{C}|_I] \geq \epsilon$. A q-query tester $\mathcal{D}$ is a (q, ϵ)-strong tester if for all $w \in \mathbb{F}^n$ we have $\mathop{\mathbf{Pr}}_{I \sim \mathcal{D}}[w|_I \notin \mathcal{C}|_I] \geq \epsilon \cdot \delta(w, \mathcal{C})$.

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 I.2. Although the tester in Definition I.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 $\mathcal{C}$.

C. Main Results

In this paper we resolve the following question raised by Goldreich and Sudan [11].

Question I.3 ([11])**.** *Are there exist constants $q \in \mathbb{N}^+$, $d, \epsilon, \gamma > 0$ and a constant size alphabet Σ such that for infinitely many $n \in \mathbb{N}^+$ we have a code $C \subseteq \Sigma^n$, where*

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

Although the requested range of parameters was achieved for the weak LTCs [9], [10], [31], strong LTCs with these parameters were not obtained and this question remained to be a basic open question in the area of LTCs.

Our main theorem (Theorem I.4) answers affirmatively on Question I.3.

Theorem I.4 (Main Theorem)**.** *There exist constants $d, \epsilon, \gamma > 0$ such that for infinitely many $n \in \mathbb{N}^+$ we have a linear code $C \subseteq \mathbb{F}_2^n$, where*

- *C is a $(3, \epsilon)$-strong LTC,*
- *$\delta(C) \geq \gamma$ and $\mathrm{rate}(C) \geq \frac{1}{\log^d n}$.*

We notice that 3 queries are necessary to test non-trivial linear codes [37].[4]

The proof of Theorem I.4 is postponed to the full version.

[3] We think that this question is pretty much known in the area, but we do not aware if it was explicitly asked in the literature.

[4] By "non-trivial" codes we mean codes with a constant relative distance and non-constant dimension.

The key ideas behind the proof of Theorem I.4.: The proof of Theorem I.4 contains three stages.

Relaxed LTCs. First, we present in Section II a new notion of *relaxed LTCs*. Intuitively, relaxed LTCs have two kind of coordinates: those with good testability and those which worse (but non-trivial) testability (see Definition II.2). Then, we present the first observation of this work (Observation II.4) and its corollary (Corollary II.5) in Section II saying that such relaxed LTCs can be easily converted to strong LTCs. Hence, all we need to resolve Question I.3 is to construct relaxed LTCs with a corresponding range of parameters.

Relaxed LTCs to start with. We want to construct sufficiently nice relaxed LTCs. To achieve the required relaxed LTCs, our starting point is the main result of [33].[5] However, we cannot use directly the codes and the testers as that were suggested in [33], i.e., they should be slightly modified before the use. So, in Section III we recall the main result of [33] and make some immediate corollaries to conclude the relaxed LTCs (with inverse polylogarithmic soundness) we will use as a starting point in the proof of Theorem I.4.

Gap Amplification can be applied to relaxed LTCs. We recall the well known "gap amplification" technique of Dinur [10] in Section IV. We show that the gap amplification and in particular, its version corresponding to linear codes [31] (see also [50]) can be applied to the linear relaxed LTCs to obtain linear relaxed LTCs with higher first soundness parameter (see Definition II.2). The crucial observation here is that while the first soundness parameter is amplified by the gap amplification procedure, the second soundness parameter of these relaxed LTCs will not be reduced too much. This observation gives us a possibility to apply the gap amplification many times and to obtain linear relaxed LTCs, where the first soundness parameter is constant and the second soundness parameter is inverse polylogarithmic. Finally, these relaxed LTCs can be converted to the strong LTCs with a constant soundness and inverse polylogarithmic rate using Corollary II.5.

1) Asymptotically Good LTCs with poly-log queries: We start this section by introducing a specific kind of junta with respect to a linear code. Intuitively, an (n', h)-junta with respect to a linear code $C \subseteq \mathbb{F}_2^n$ is a junta of size n' such that every code symbol outside this junta is determined by at most h code symbols of the junta.

Definition I.5 (Junta)**.** Let $C \subseteq \mathbb{F}_2^n$ be a linear code and $T \subseteq [n]$ be a subset. We say that T is an (n', h)-junta with respect to C if $n' = |T|$ and for every $j \in [n] \setminus T$ there exists $u_j \in C^\perp$ such that $j \in \mathrm{supp}(u)$, $\mathrm{supp}(u_j) \setminus \{j\} \subseteq T$ and $|\mathrm{supp}(u_j) \setminus \{j\}| \leq h$.

We notice that every linear code C has a $(\dim(C), \dim(C))$-junta. To see this assume without loss of generality that the generating matrix of C has a systematic form[6] and let $T = [\dim(C)]$.

Recall that Theorem I.4 shows the existence of strong LTCs with constant query complexity, soundness, relative distance and inverse polylogarithmic rate. The following conjecture argues that strong LTCs with poly-log query complexity and inverse poly-log rate can be accompanied with a $(O(\dim(C)), \mathrm{polylog}(n))$-junta.

Conjecture I.6 (strong LTC with a junta)**.** *There exists a linear code $C \subseteq \mathbb{F}_2^n$ (for arbitrary large $n \in \mathbb{N}^+$) such that C is a $(\mathrm{polylog}(n), \frac{1}{2})$-strong LTC, $\delta(C) = \Omega(1)$, $\mathrm{rate}(C) \geq \frac{1}{\mathrm{polylog}(n)}$ and a $(\Theta(\dim(C)), \mathrm{polylog}(n))$-junta T with respect to C.*

Remark I.7. The construction of strong LTCs presented in [33, Corollary 3.2] seems close to resolve Conjecture I.6, but doesn't resolve it. Informally, this construction was obtained by execution $\Theta(\log\log(n))$ iterations (see Remark III.2) and gave a $(\mathrm{polylog}(n), \frac{1}{2})$-strong LTC C with $\delta(C) = \Omega(1)$ and $\mathrm{rate}(C) \geq \frac{1}{\mathrm{polylog}(n)}$. Each iteration 3 procedures were applied: the star product, the distance amplification and the random projection. A natural candidate for a $(\Theta(\dim(C)), \mathrm{polylog}(n))$-junta would be the core $A(C)$ of the code $C \subseteq \mathbb{F}_2^n$ constructed in [33], which had a size $|A(C)| = \Theta(\dim(C))$, i.e., $\mathrm{blocklength}(C|_{A(C)}) = \Theta(\dim(C))$. The problem is that only 2 procedures: the star product and the distance amplification preserved the required property, i.e., there exists a fixed constant $r \in \mathbb{N}^+$ such that if a core $A(C)$ of the input code C is a $(\Theta(\dim(C)), h)$-junta then the core $A(C')$ of the code C' obtained by these procedures is a $(\Theta(\dim(C)), r \cdot h)$-junta. Unfortunately, the random projection procedure does not preserve this property. However, if there exists a way to make this procedure preserving the "junta" property as another two procedures, then after the execution of $\Theta(\log\log(n))$ iterations we would get not only a $(\mathrm{polylog}(n), \frac{1}{2})$-strong LTC C, but also a $(\Theta(\dim(C)), \mathrm{polylog}(n))$-junta $A(C)$.

Under Conjecture I.6 it is not hard to prove the existence of asymptotically good strong LTCs with polylogarithmic query complexity.

Theorem I.8 (Asymptotically good LTCs with poly-log queries)**.** *Under Conjecture I.6, there exists a linear $(\mathrm{polylog}(n'), \frac{1}{2})$-strong LTC $C' \subseteq \mathbb{F}_2^{n'}$ (for arbitrary large n') such that $\delta(C') = \Omega(1)$ and $\mathrm{rate}(C') = \Omega(1)$.*

We stress that the corresponding statement of Theorem I.8 is open even for weak LTCs. The proof of Theorem I.8 is postponed to the full version.

[5]The codes presented in [33] were very similar to the codes of [31].

[6]Such generating matrix yields codewords whose first $\dim(C)$ symbols are message symbols.

II. Relaxed LTCs

Before we present Observation II.4, we recall some concept used in [33].

Definition II.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 $\delta(w|_{[n]\setminus A(C)}, w'|_{[n]\setminus A(C)}) = \delta(w|_{[n]\setminus A(C)}, C|_{[n]\setminus A(C)}) = 0$.

Our first novelty is the following concept of a relaxed LTC (rLTC).

Definition II.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 $\Pr_{I\sim\mathcal{D}}[w|_I \notin C|_I] \geq \max\left\{\epsilon_1 \cdot \delta(w|_{A(C)}, c|_{A(C)}), \epsilon_2 \cdot \delta(w|_{-A(C)}, c|_{-A(C)})\right\}$. 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.

Remark II.3. We note that if $C \subseteq \mathbb{F}^n$ is a (q, ϵ)-strong LTC and $\mathcal{D}$ is its tester, then setting $A(C) = [n]$ it holds that C is a $(q, \epsilon, 1)$-rLTC with regards to the same tester $\mathcal{D}$ because for every $w \in \mathbb{F}^n$ we have

$$\Pr_{I\sim\mathcal{D}}[w|_I \notin C|_I] \geq \epsilon \cdot \delta(w, C) = \max\left\{\epsilon \cdot \delta(w|_{[n]}, C|_{[n]}), 1 \cdot 0\right\} =$$

$$= \max\left\{\epsilon \cdot \delta(w|_{A(C)}, C|_{A(C)}), 1 \cdot \delta(w|_{-A(C)}, C|_{-A(C)})\right\}.$$

Our first observation in this work is that a relaxed LTC with sub-constant second soundness parameter can be easily converted to a strong LTC with a constant soundness.

Observation II.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 $O(n')$.*

The proof of Observation II.4 is omitted due to the space limitations.

Although Observation II.4 might seem naive, it implies the following corollary that will play a crucial role in the proof of Theorem I.4.

Corollary II.5. *Assume that for constants $q \geq 2, \epsilon > 0$ and infinitely many $n \in \mathbb{N}^+$ we have a linear code $C \subseteq \mathbb{F}_2^n$ 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 $C' \subseteq \mathbb{F}_2^{n'}$ such that $n \leq n' \leq n \cdot \mathrm{polylog}(n)$, C' is a $(q, \epsilon/6)$-strong LTC, $\delta(C') = \Omega(1)$ and $\mathrm{rate}(C') = \frac{1}{\mathrm{polylog}(n')}$ (i.e., Question I.3 is solved).*

III. The main result of [33] and its corollaries

In this section we recall the main result of [33]. Then we make some corollaries that will be used later.

Theorem III.1 ([33])**.** *For some constant $d \in \mathbb{N}^+$ and infinitely many $n \in \mathbb{N}^+$ there exists a linear code $C \subseteq \mathbb{F}_2^n$ and its tester $\mathcal{D}$ such that*

- *C is a $(3, \frac{1}{\log^d n})$-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 $|N_{\mathcal{D}}(i)| \leq \log^d n$.*

Remark III.2. Although in [33] two last bullets were not proved, but one could verify that these bullets hold. The construction in [33] begins from a constant blocklength code C_1 and contained 3 procedures: the star product, the random projection and the distance amplification. These 3 procedures were applied iteratively $\Theta(\log\log n)$ times. Each iteration i is executed on the code C_i that had a blocklength n_i and a tester $\mathcal{D}_i$. The output of each iteration i is the code C_{i+1}.

Initially, the base code $C_1 \subseteq \mathbb{F}_2^{n_1}$ and its tester $\mathcal{D}_1$ satisfied the last two bullets with respect to its blocklength $n_1 = O(1)$. I.e., $|\mathrm{supp}(\mathcal{D}_1)| \leq n_1 \cdot O(1)$ and for every $u \in \mathrm{supp}(\mathcal{D}_1)$ it holds that $\mathcal{D}_1(u) \leq \frac{O(1)}{n_1}$. Moreover, for every $i \in [n_1]$ we have $|N_{\mathcal{D}_1}(i)| \leq O(1)$.

Each iteration, the star product, the random projection and the distance amplification procedures were applied. The random projection does not affect the properties listed in these bullets, but only rearranges the coordinates of the given code in some way. The star product and the distance amplification procedure do affect the properties listed in these bullets, but only by fixed multiplicative constants.

More formally, there exists a fixed constant $h > 0$ such that the following occur. Suppose that in the iteration i for some $h_i > 0$ we have the code $C_i \subseteq \mathbb{F}^{n_i}$ and its tester $\mathcal{D}_i$ such that $\mathrm{supp}(\mathcal{D}_i) \leq h_i \cdot n_i$, for every $u \in \mathrm{supp}(\mathcal{D}_i)$ it holds that $\mathcal{D}_i(u) \leq \frac{h_i}{n_i}$, and for every $j \in [n_i]$ it holds that $|N_{\mathcal{D}_i}(j)| \leq h_i$. Then, after the star product (or distance amplification) is applied, resulting in the code $C_{i+1} \subseteq \mathbb{F}^{n_{i+1}}$

and its tester $\mathcal{D}_{i+1}$, we have $\mathrm{supp}(\mathcal{D}_{i+1}) \leq h \cdot h_i \cdot n_{i+1}$, for every $u \in \mathrm{supp}(\mathcal{D}_{i+1})$ it holds that $\mathcal{D}_{i+1}(u) \leq \frac{h \cdot h_i}{n_{i+1}}$, and for every $j \in [n_{i+1}]$ it holds that $|N_{\mathcal{D}_{i+1}}(j)| \leq h \cdot h_i$.

Therefore, after $\Theta(\log \log n)$ iterations we obtain the code $C \subseteq \mathbb{F}^n$ and its tester $\mathcal{D}$ such that $|\mathrm{supp}(\mathcal{D})| \leq n \cdot \mathrm{polylog}(n)$, for every $u \in \mathrm{supp}(\mathcal{D})$ it holds that $\mathcal{D}(u) \leq \frac{\mathrm{polylog}(n)}{n}$, and for every $j \in [n]$ it holds that $|N_{\mathcal{D}}(j)| \leq \mathrm{polylog}(n)$.

We pay attention that one can turn the strong LTCs of Theorem III.1 to the strong LTCs with a uniform distribution over the tests, and the soundness parameter, roughly speaking, will be preserved.

Corollary III.3. *For some constant $d \in \mathbb{N}^+$ and infinitely many $n \in \mathbb{N}^+$ there exist a linear code $C' \subseteq \mathbb{F}_2^n$ and its tester $\mathcal{D}'$ which is a uniform distribution over $\mathrm{supp}(\mathcal{D}')$ such that*

- *C' is a $(3, \frac{1}{\log^d n})$-strong LTC with respect to $\mathcal{D}'$,*
- *$\delta(C') = \Omega(1)$,*
- *$\mathrm{rate}(C') \geq \frac{1}{\log^d n}$,*
- *$|\mathrm{supp}(\mathcal{D}')| \leq n \log^d n$, and*
- *for every $i \in [n]$ we have $|N_{\mathcal{D}'}(i)| \leq \log^d n$.*

Now, in Corollary III.4 we show that the 3-query strong LTCs over $\mathbb{F}_2$ from Corollary III.3 can be easily converted to the 2-query rLTCs over $\mathbb{F}_2^3$ with a similar range of parameters.[7] This conversion is standard (for the case of LTCs, PCPs and assignment testers) and was explained, e.g., in [10], [31].

Corollary III.4. *For some constant $d \in \mathbb{N}^+$ and infinitely many $n'' \in \mathbb{N}^+$ there exist a code $C'' \subseteq (\mathbb{F}_2^3)^{(n'')}$ and its tester $\mathcal{D}''$ which is a uniform distribution over $\mathrm{supp}(\mathcal{D}'')$ such that*

- *C'' is linear over $\mathbb{F}_2$,*
- *C'' is a $(2, \frac{1}{3\log^d n''}, \frac{1}{6\log^{2d} n''})$-rLTC with respect to its core $A(C'')$ and $\mathcal{D}''$,*
- *$\delta(C''|_{A(C'')}) = \Omega(1)$,*
- *$\mathrm{rate}(C'') \geq \frac{1}{2\log^{2d} n''}$,*
- *$|\mathrm{supp}(\mathcal{D}'')| \leq 3 \cdot n''$, and*
- *for every $i \in [n'']$ we have $|N_{\mathcal{D}''}(i)| \leq \log^d n''$.*

IV. Gap Amplification Procedure for LTCs

In this section we describe the main result of Dinur [10] and its affect on the locally testable codes. We notice that two interesting alternatives were proposed. Radhakrishnan [51] suggested another option for the amplification lemma [10], where he used lazy random walks in the constraints graph. In particular, this suggestion improves some of the constants inside the Dinur's results. Goldreich and Meir [52] pointed out on a small gap in the proof of the amplification of assignment testers in [10]. Namely, while Dinur [10] argued that every an execution of the gap amplification costs a linear blowup for the underlying graph size, Goldreich and Meir [52] showed that sometimes this blowup can be larger and showed how one can easily correct this to have always only a linear blowup.

Nevertheless, in our paper we don't need the mentioned suggestions/corrections and we address the original work of Dinur [10]. The only modification we need is that the linearity of the underlying code can be preserved if the alphabet reduction stage in the gap amplification will be done by the concatenation with the Hadamard code [31, Section 6.4.3], and not by a general assignment testers composition as in [10]. Now we recall the gap amplification procedure [10] and describe how it is applied on the linear codes, and in particular to the 2-query linear relaxed LTCs.

A 2-query LTC can be associated with a constraints graph.: In this section let $\mathbb{F} = \mathbb{F}_2^3$. Assume $C \subseteq \mathbb{F}^n$ has a 2-query tester $\mathcal{D}$. Let $G = (V, E)$ be an undirected graph, where $V = [n]$ and $\{i, j\} \in E$ if and only if $\mathcal{D}(\{i, j\}) > 0$. The degree of a symbol i of the code C is associated to the degree of the node i in the graph G, and equal to $|\{j \in [n] \mid \mathcal{D}(\{i, j\}) > 0\}|$.

The gap amplification for a relaxed LTC.: Let us recall how the gap amplification would be applied on a relaxed LTC. This will be almost identical to the execution of the gap amplification on the assignment testers [10], where a single modification is that the alphabet reduction is done by the concatenation with the Hadamard code as was explained in [31, Section 6.4.3]. Assume that the input is the relaxed LTC $C \subseteq \mathbb{F}^n$ and its 2-query tester $\mathcal{D}$ such that $\mathcal{D}$ is uniform over $\mathrm{supp}(\mathcal{D})$. Assume that $A(C) \subseteq [n]$ is the core of the code C and without loss of generality assume that $A(C) = [|A(C)|]$, i.e., the core of the code is the first $|A(C)|$ coordinates. It is important to note that during the execution of the gap amplification the symbols of the core will be preserved in every stage of this procedure.

Let G be the graph corresponding to the code C and its 2-query tester $\mathcal{D}$. The gap amplification procedure contains the following three stages.

First stage - Preprocessing (described in [10, Section 4]) Assume that a coordinate i in C has degree d_i with respect to the tester $\mathcal{D}$. Let $d = \max_{i \in [n]} d_i$. The code $C \subseteq \mathbb{F}^n$ and its tester $\mathcal{D}$ are transformed to the new code $C' \subseteq \mathbb{F}^{n'}$ and its tester $\mathcal{D}'$ such that $n \leq n' \leq d \cdot n$ and $C'|_{[n]} = C$, i.e., all old code entries are preserved and some new code entries are added. It also holds that $\mathcal{D}'$ is uniform over a new collection of 2-query constraints, i.e., $\mathcal{D}'$ is uniform over $\mathrm{supp}(\mathcal{D}')$. The new entries are added by duplicating some original entries. The number of 2-query tests in $\mathcal{D}'$ is $|\mathrm{supp}(\mathcal{D}')| = O(d \cdot |\mathrm{supp}(\mathcal{D})|)$. The degree of every index i in the code C' with respect to the tester $\mathcal{D}'$ is a fixed constant (independent of any parameters). If the code C' is associated with a graph G', then G' is a constant degree expander graph (see [10, Section 4]).

[7] During this paper we associate $\mathbb{F}_2^3$ with $\mathbb{F}_{2^3}$.

We set the core of the code C' to be $A(C') = A(C)$ and note that $C|_{A(C)} = C'|_{A(C')}$, i.e., the core symbols are preserved. Similarly to the proof presented by Dinur [10], one could verify that if for every $w \in \mathbb{F}^n$ it holds that $\mathbf{Pr}_{I\sim\mathcal{D}}[w|_I \notin C|_I] \geq \epsilon_1 \cdot \delta(w|_{A(C)}, C|_{A(C)})$, then for every $w \in \mathbb{F}^{n'}$ it holds that $\mathbf{Pr}_{I\sim\mathcal{D}'}[w|_I \notin C'|_I] \geq (\epsilon_1/d) \cdot \delta(w|_{A(C')}, C'|_{A(C')})$. In particular, this means that if C is a $(2, \epsilon_1, \cdot)$-rLTC with respect to its tester $\mathcal{D}$ and its core $A(C)$, then C' is a $(2, \epsilon_1/d, \cdot)$-rLTC with respect to its tester $\mathcal{D}'$ and its core $A(C')$. I.e., the decrease in the first soundness parameter is bounded by a maximal degree of a code coordinate.

Notice that if all degrees of the code coordinates are upper-bounded by a fixed constant, then the soundness will be decreased be only by a constant.

Second Stage - Amplification (described in [10, Sections 1 and 6]) The input of this stage is the code $C' \subseteq \mathbb{F}^{n'}$ and its 2-query tester $\mathcal{D}'$ which is uniform over its support. We recall that $A(C') \subseteq [n]$ is a core of the code C' such that $A(C') = [|A(C')|]$. We also know that the degree of every index of C' (after the first stage) with respect to $\mathcal{D}'$ is equal to a fixed constant $d \in \mathbb{N}^+$. In this stage we associate the code C' and the tester $\mathcal{D}'$ with a graph G'. Then the graph G' is transformed to the graph $(G')^t$ for sufficiently large constant $t \in \mathbb{N}^+$, where $(G')^t$ has the same vertexes as G' and the edge set $(E')^t$ contains k parallel edges between i_1 and i_2 if and only if the number of t-step walks from i_1 to i_2 is exactly k. The graph $(G')^t$ defines the code C'' and its tester $\mathcal{D}''$ which is uniform over all edges of the graph $(G')^t$, but the first $|A(C')|$ symbols are exactly the first $|A(C')|$ symbols of C', i.e., the core coordinates are preserved. We set $A(C'') = A(C')$. The underlying field of the code C'' is $\mathbb{F}^{d^{\lceil t/2 \rceil}}$, but the symbols indexed by $A(C'')$ belong to $\mathbb{F}$. Note that C'' is linear over $\mathbb{F}$. In particular, the blocklength of C'' is n' and $|\operatorname{supp}(\mathcal{D}'')| \leq |\operatorname{supp}(\mathcal{D}')| \cdot d^t$.

In [10] it is shown that the first soundness parameter is increased in $t' = \Omega(\sqrt{(t)})$, where the constant inside $\Omega(\cdot)$ is independent of t, and hence t is picked to be sufficiently large constant such that, e.g., $t' \geq 10$. As we mentioned, Radhakrishnan [51] improved the dependency on t, but we don't use his result in this paper.

That means if for every $w \in \mathbb{F}^{n'}$ it holds that $\mathbf{Pr}_{I\sim\mathcal{D}'}[w|_I \notin C'|_I] \geq \epsilon_1 \cdot \delta(w|_{A(C')}, C|_{A(C')})$, then for every $w \in \mathbb{F}^{n''}$ it holds that $\mathbf{Pr}_{I\sim\mathcal{D}''}[w|_I \notin C''|_I] \geq (\epsilon_1 \cdot t') \cdot \delta(w|_{A(C'')}, C'|_{A(C'')})$. Namely, if C' was a $(2, \epsilon_1, \cdot)$-rLTC with respect to $A(C')$ and $\mathcal{D}'$ then C'' is a $(2, t' \cdot \epsilon_1, \cdot)$-rLTC with respect to $A(C'')$ and $\mathcal{D}''$, where ϵ_1 is less than some fixed constant $\gamma > 0$.

Third Stage - Alphabet Reduction (described in [10, Sections 1 and 5]) and [31, Section 6.4.3] In this stage, we will use the suggestion of Meir [31, Section 6.4.3], where the alphabet reduction is done by the concatenation with the binary Hadamard code. In this stage, every code symbol, which is an element of $\mathbb{F}^{d^t}$ for some $d, t \in \mathbb{N}^+$, is encoded by the Hadamard code over the field $\mathbb{F}_2$. The output of this stage is a code $C''' \subseteq \mathbb{F}^{n'''}$ and its 2-query tester $\mathcal{D}'''$ which is uniform over its support. The core of the code is preserved again, and we set $A(C''') = A(C'')$ and note that $C''|_{A(C'')} = C'''|_{A(C''')}$.

As was explained in [31, Section 6.4.3], this reduction decreases rejection probability by a fixed constant $g > 0$ (independent of the parameters of the code), i.e., if for every $w \in \mathbb{F}^{n''}$ it holds that $\mathbf{Pr}_{I\sim\mathcal{D}''}[w|_I \notin C''|_I] \geq \epsilon_1 \cdot \delta(w|_{A(C'')}, C|_{A(C'')})$, then for every $w \in \mathbb{F}^{n'''}$ it holds that $\mathbf{Pr}_{I\sim\mathcal{D}'''}[w|_I \notin C'''|_I] \geq (\epsilon_1/g) \cdot \delta(w|_{A(C''')}, C'|_{A(C''')})$. Namely, if C'' was a $(2, \epsilon_1, \cdot)$-rLTC with respect to $A(C'')$ and $\mathcal{D}''$ then C''' is a $(2, \epsilon_1/g, \cdot)$-rLTC with respect to $A(C''')$ and $\mathcal{D}'''$.

An interesting point is that there are two options:

1) to obtain the binary linear code C''', where $\mathcal{D}'''$ is a 3-query tester. As was said, this is done simply by the concatenation with the binary Hadamard code (see [31, Section 6.4.3]).
2) to obtain the code C''' over the field $\mathbb{F}$, where $\mathcal{D}'''$ is a 2-query tester. This can be done by applying the first bullet and then turn the 3-query rLTC over $\mathbb{F}_2$ to the 2-query rLTC over $\mathbb{F} = \mathbb{F}_2^3$ using the standard technique (as in Corollary III.4).

This gap amplification procedure will be applied a number of times. Each iteration, besides the last one, we use the second bullet, i.e., we obtain a 2-query rLTC over $\mathbb{F}$ (which is linear over $\mathbb{F}_2$). This code can be passed to the new iteration of gap amplification. However, in the last iteration we choose the first bullet and obtain a binary linear 3-query rLTC.

Overall, the output of the gap amplification procedure is the code C''' and its tester $\mathcal{D}'''$.

V. Open Questions and Discussions

This work leaves two open questions. The first one is obtaining asymptotically good strong LTCs with poly-log query complexity and constant soundness. In Theorem I.8 we argued their existence under Conjecture I.6. One can try to prove this conjecture. Our feel is that it might be possible to implement the random projection operation [31], [33] to preserve the "junta" property (see Remark I.7). E.g., it might be possible to argue that there exists some invariant feature that is preserved each iteration in the construction of [33] and use this feature to re-implement the random projection operation. Resolving this task would yield an unconditional proof for Theorem I.8.

The second open question, mentioned in [11], [31], is the *explicit* construction of strong LTCs with inverse poly-logarithmic rate, constant relative distance, constant query complexity and constant soundness. Recall that our construction of strong LTCs in Theorem I.4 is based on the construction of [33] (which is almost identical to [31]),

where the construction of [33] was probabilistic. One of possible approach to provide an explicit construction of such strong LTCs is by applying the arguments of [33] and the arguments used in this paper to the construction of Ben-Sasson and Sudan [9]. While the work [9] yields weak LTCs, the underlying construction has some similarities to the constructions of [31], [33] discussed [31, Sectioin 7.2]. On the other hand, the ideas presented in [33] seem fairly general and it might that these ideas can be applied to [9] to conclude the explicit construction of strong LTCs with the required range of parameters.

ACKNOWLEDGMENT

The author thanks Eli Ben-Sasson for helpful discussions. We would like to thank Or Meir for raising the suggestion (discussed in Section V) to get an explicit construction of strong LTCs by applying the arguments of [33] to the codes of [9].

The research has received funding from the European Research Council as part of the ERC project CaC (grant 259426).

REFERENCES

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, "Proof Verification and the Hardness of Approximation Problems," *Journal of the ACM*, vol. 45, no. 3, pp. 501–555, May 1998.

[2] S. Arora and S. Safra, "Probabilistic Checking of Proofs: A New Characterization of NP," *Journal of the ACM*, vol. 45, no. 1, pp. 70–122, Jan. 1998.

[3] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy, "Interactive proofs and the hardness of approximating cliques," *Journal of the ACM*, vol. 43, no. 2, pp. 268–292, 1996. [Online]. Available: http://doi.acm.org/10.1145/226643.226652

[4] L. Babai, L. Fortnow, L. A. Levin, and M. 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*. ACM, 1991, pp. 21–31. [Online]. Available: http://doi.acm.org/10.1145/103418.103428

[5] M. Bellare, O. Goldreich, and M. Sudan, "Free Bits, PCPs, and Nonapproximability—Towards Tight Results," *SIAM Journal on Computing*, vol. 27, no. 3, pp. 804–915, Jun. 1998.

[6] 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)*, ACM SIGACT. New York: ACM Press, 1993, pp. 294–304.

[7] J. Håstad, "Some optimal inapproximability results," *Journal of the ACM*, vol. 48, no. 4, pp. 798–859, 2001. [Online]. Available: http://doi.acm.org/10.1145/502090.502098

[8] E. Ben-Sasson, O. Goldreich, P. Harsha, M. Sudan, and S. P. Vadhan, "Robust PCPs of Proximity, Shorter PCPs, and Applications to Coding," *SIAM Journal on Computing*, vol. 36, no. 4, pp. 889–974, 2006.

[9] E. Ben-Sasson and M. Sudan, "Short PCPs with polylog query complexity," *SIAM J. Comput*, vol. 38, no. 2, pp. 551–607, 2008. [Online]. Available: http://dx.doi.org/10.1137/050646445

[10] I. Dinur, "The PCP theorem by gap amplification," *Journal of the ACM*, vol. 54, no. 3, pp. 12:1–12:44, Jun. 2007.

[11] O. Goldreich and M. Sudan, "Locally testable codes and PCPs of almost-linear length," *Journal of the ACM*, vol. 53, no. 4, pp. 558–655, Jul. 2006.

[12] L. Trevisan, "Some Applications of Coding Theory in Computational Complexity," Sep. 23 2004. [Online]. Available: http://arxiv.org/abs/cs/0409044

[13] O. Goldreich, "Short Locally Testable Codes and Proofs (Survey)," *Electronic Colloquium on Computational Complexity (ECCC)*, no. 014, 2005. [Online]. Available: http://eccc.hpi-web.de/eccc-reports/2005/TR05-014/index.html

[14] N. Alon, T. Kaufman, M. Krivelevich, S. Litsyn, and D. Ron, "Testing Reed-Muller codes," *IEEE Transactions on Information Theory*, vol. 51, no. 11, pp. 4032–4039, 2005. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TIT.2005.856958

[15] M. Blum, M. Luby, and R. Rubinfeld, "Self-testing/correcting with applications to numerical problems," *Journal of Computer and System Sciences*, vol. 47, no. 3, pp. 549–595, Dec. 1993.

[16] E. Ben-Sasson, N. Ron-Zewi, and M. Sudan, "Sparse affine-invariant linear codes are locally testable," in *FOCS*. IEEE Computer Society, 2012, pp. 561–570. [Online]. Available: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6374356

[17] E. Ben-Sasson, E. Grigorescu, G. Maatouk, A. Shpilka, and M. Sudan, "On sums of locally testable affine invariant properties," in *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, ser. Lecture Notes in Computer Science, vol. 6845. Springer, 2011, pp. 400–411. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-22935-0

[18] A. Bhattacharyya, S. Kopparty, G. Schoenebeck, M. Sudan, and D. Zuckerman, "Optimal testing of reed-muller codes," in *FOCS*. IEEE Computer Society, 2010, pp. 488–497. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/FOCS.2010.54

[19] E. Grigorescu, T. Kaufman, and M. Sudan, "Succinct representation of codes with applications to testing," *SIAM J. Discrete Math*, vol. 26, no. 4, pp. 1618–1634, 2012. [Online]. Available: http://dx.doi.org/10.1137/100818364

[20] T. Kaufman and D. Ron, "Testing polynomials over general fields," *SIAM J. Comput*, vol. 36, no. 3, pp. 779–802, 2006. [Online]. Available: http://dx.doi.org/10.1137/S0097539704445615

[21] T. Kaufman and M. 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*. ACM, 2008, pp. 403–412. [Online]. Available: http://doi.acm.org/10.1145/1374376.1374434

[22] T. Kaufman and S. 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)*. IEEE, 2011, pp. 788–796. [Online]. Available: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6108120

[23] N. Ron-Zewi and M. 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)*, ser. Lecture Notes in Computer Science, vol. 7408. Springer, 2012, pp. 639–650. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-32512-0

[24] I. Dinur and O. Reingold, "Assignment Testers: Towards a Combinatorial Proof of the PCP Theorem," *SIAM Journal on Computing*, vol. 36, no. 4, pp. 975–1024, 2006. [Online]. Available: http://dx.doi.org/10.1137/S0097539705446962

[25] E. Ben-Sasson and M. Viderman, "Low rate is insufficient for local testability," in *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, ser. Lecture Notes in Computer Science, vol. 6302. Springer, 2010, pp. 420–433. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15369-3

[26] T. Kaufman and M. Sudan, "Sparse Random Linear Codes are Locally Decodable and Testable," in *FOCS*. IEEE Computer Society, 2007, pp. 590–600. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/FOCS.2007.65

[27] S. Kopparty and S. 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)*. ACM, 2010, pp. 417–426. [Online]. Available: http://doi.acm.org/10.1145/1806689.1806748

[28] I. Dinur, M. Sudan, and A. Wigderson, "Robust Local Testability of Tensor Products of LDPC Codes," in *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, ser. Lecture Notes in Computer Science, vol. 4110. Springer, 2006, pp. 304–315. [Online]. Available: http://dx.doi.org/10.1007/11830924_29

[29] E. Ben-Sasson and M. Viderman, "Tensor Products of Weakly Smooth Codes are Robust," *Theory of Computing*, vol. 5, no. 1, pp. 239–255, 2009. [Online]. Available: http://dx.doi.org/10.4086/toc.2009.v005a012

[30] ——, "Composition of Semi-LTCs by Two-Wise Tensor Products," in *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, ser. Lecture Notes in Computer Science, vol. 5687. Springer, 2009, pp. 378–391. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03685-9

[31] O. Meir, "Combinatorial Construction of Locally Testable Codes," *SIAM J. Comput*, vol. 39, no. 2, pp. 491–544, 2009. [Online]. Available: http://dx.doi.org/10.1137/080729967

[32] M. Viderman, "A combination of testability and decodability by tensor products," in *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, ser. Lecture Notes in Computer Science, vol. 7408. Springer, 2012, pp. 651–662. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-32512-0

[33] ——, "Strong LTCs with inverse polylogarithmic rate and soundness," *To appear in CCC 2013. Electronic Colloquium on Computational Complexity (ECCC)*, vol. 19, p. 168, 2012. [Online]. Available: http://eccc.hpi-web.de/report/2012/168

[34] O. Goldreich, "Home page." [Online]. Available: http://www.wisdom.weizmann.ac.il/~oded/;http://dl.acm.org/author_page.cfm?id=81336489395

[35] O. Goldreich and D. Ron, "On proximity-oblivious testing," *SIAM J. Comput*, vol. 40, no. 2, pp. 534–566, 2011. [Online]. Available: http://dx.doi.org/10.1137/100789646

[36] E. Ben-Sasson and M. Viderman, "Towards lower bounds on locally testable codes via density arguments," *Computational Complexity*, vol. 21, no. 2, pp. 267–309, 2012. [Online]. Available: http://dx.doi.org/10.1007/s00037-012-0042-8

[37] E. Ben-Sasson, O. Goldreich, and M. Sudan, "Bounds on 2-Query Codeword Testing," in *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, ser. Lecture Notes in Computer Science, vol. 2764. Springer, 2003, pp. 216–227. [Online]. Available: http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2764&spage=216

[38] V. Guruswami, "On 2-Query Codeword Testing with Near-Perfect Completeness," in *Proceedings of the 17th International Symposium on Algorithms and Computation (ISAAC)*, ser. Lecture Notes in Computer Science, vol. 4288. Springer, 2006, pp. 267–276. [Online]. Available: http://dx.doi.org/10.1007/11940128_28

[39] G. Kol and R. Raz, "Locally testable codes analogues to the unique games conjecture do not exist," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 16, p. 128, 2009.

[40] ——, "Bounds on 2-query locally testable codes with affine tests," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 16, p. 138, 2009.

[41] E. Ben-Sasson, P. Harsha, and S. Raskhodnikova, "Some 3CNF Properties Are Hard to Test," *SIAM Journal on Computing*, vol. 35, no. 1, pp. 1–21, 2005. [Online]. Available: http://epubs.siam.org/SICOMP/volume-35/art_44544.html

[42] L. Babai, A. Shpilka, and D. Stefankovic, "Locally testable cyclic codes," in *Proceedings: 44th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2003, 11–14 October 2003, Cambridge, Massachusetts*. IEEE Computer Society Press, 2003, pp. 116–125.

[43] E. Ben-Sasson, G. Maatouk, A. Shpilka, and M. Sudan, "Symmetric LDPC Codes are not Necessarily Locally Testable," in *IEEE Conference on Computational Complexity*. IEEE Computer Society, 2011, pp. 55–65. [Online]. Available: http://dx.doi.org/10.1109/CCC.2011.14

[44] E. Ben-Sasson and M. Sudan, "Limits on the Rate of Locally Testable Affine-Invariant Codes," in *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, ser. Lecture Notes in Computer Science, vol. 6845. Springer, 2011, pp. 412–423. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-22935-0

[45] E. Grigorescu, T. Kaufman, and M. Sudan, "2-Transitivity Is Insufficient for Local Testability," in *IEEE Conference on Computational Complexity*. IEEE Computer Society, 2008, pp. 259–267. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/CCC.2008.31

[46] T. Kaufman and A. Wigderson, "Symmetric LDPC codes and local testing," in *Innovations in Computer Science - ICS, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, A. C.-C. Yao, Ed. Tsinghua University Press, 2010, pp. 406–421. [Online]. Available: http://conference.itcs.tsinghua.edu.cn/ICS2010/content/papers/32.html

[47] E. Ben-Sasson, V. Guruswami, T. Kaufman, M. Sudan, and M. Viderman, "Locally Testable Codes Require Redundant Testers," *SIAM J. Comput*, vol. 39, no. 7, pp. 3230–3247, 2010. [Online]. Available: http://dx.doi.org/10.1137/090779875

[48] I. Dinur and T. Kaufman, "Dense locally testable codes cannot have constant rate and distance," in *Proceedings of Approximation, Randomization, and Combinatorial Optimization (APPROX-RANDOM)*, vol. 6845, 2011, pp. 507–518. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-22935-0

[49] E. Ben-Sasson and M. Sudan, "Robust locally testable codes and products of codes," *Random Struct. Algorithms*, vol. 28, no. 4, pp. 387–402, 2006. [Online]. Available: http://dx.doi.org/10.1002/rsa.20120

[50] E. Ben-Sasson, P. Harsha, O. Lachish, and A. Matsliah, "Sound 3-Query PCPPs Are Long," *TOCT*, vol. 1, no. 2, 2009. [Online]. Available: http://doi.acm.org/10.1145/1595391.1595394

[51] J. Radhakrishnan, "Gap amplification in PCPs using lazy random walks," in *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part I*, ser. Lecture Notes in Computer Science, vol. 4051. Springer, 2006, pp. 96–107. [Online]. Available: http://dx.doi.org/10.1007/11786986_10

[52] O. Goldreich and O. Meir, "A Small Gap in the Gap Amplification of Assignment Testers," *Electronic Colloquium on Computational Complexity (ECCC) - TR05-046. Comment 3*, 2007. [Online]. Available: http://eccc.hpi-web.de/eccc-reports/2005/TR05-046/index.html

[53] M. Bellare and M. Sudan, "Improved non-approximability results," in *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing (STOC), 23-25 May 1994, Montréal, Québec, Canada*. ACM, 1994, pp. 184–193. [Online]. Available: http://doi.acm.org/10.1145/195058.195129

[54] M. Bellare, D. Coppersmith, J. Håstad, M. A. Kiwi, and M. Sudan, "Linearity testing in characteristic two," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1781–1795, 1996. [Online]. Available: http://dx.doi.org/10.1109/18.556674

[55] T. Kaufman, S. Litsyn, and N. Xie, "Breaking the epsilon-soundness bound of the linearity test over GF(2)," *SIAM J. Comput*, vol. 39, no. 5, pp. 1988–2003, 2010. [Online]. Available: http://dx.doi.org/10.1137/080715548