Local Testing for Membership in Lattices

Karthekeyan Chandrasekaran[1], Mahdi Cheraghchi[2], Venkata Gandikota[3], and Elena Grigorescu[4]

1 **University of Illinois, Urbana-Champaign, IL, USA**
karthe@illinois.edu
2 **Imperial College London, UK**
cheraghchi@berkeley.edu
3 **Purdue University, West Lafayette, IN, USA**
vgandiko@purdue.edu
4 **Purdue University, West Lafayette, IN, USA**
elena-g@purdue.edu

Abstract

Testing membership in lattices is of practical relevance, with applications to integer programming, error detection in lattice-based communication and cryptography. In this work, we initiate a systematic study of *local testing* for membership in lattices, complementing and building upon the extensive body of work on locally testable codes. In particular, we formally define the notion of local tests for lattices and present the following:

1. We show that in order to achieve low query complexity, it is sufficient to design one-sided non-adaptive *canonical* tests. This result is akin to, and based on an analogous result for error-correcting codes due to Ben-Sasson *et al.* (SIAM J. Computing 35(1) pp1–21).
2. We demonstrate upper and lower bounds on the query complexity of local testing for membership in *code formula* lattices. We instantiate our results for code formula lattices constructed from Reed-Muller codes to obtain nearly-matching upper and lower bounds on the query complexity of testing such lattices.
3. We contrast lattice testing from code testing by showing lower bounds on the query complexity of testing low-dimensional lattices. This illustrates large lower bounds on the query complexity of testing membership in *knapsack lattices*. On the other hand, we show that knapsack lattices with bounded coefficients have low-query testers if the inputs are promised to lie in the span of the lattice.

1998 ACM Subject Classification "F.2 Analysis of algorithms and problem complexity"

Keywords and phrases Lattices, Property Testing, Locally Testable Codes, Complexity Theory

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2016.

1 Introduction

Local testing for properties of combinatorial and algebraic objects have widespread applications and have been intensely investigated in the past few decades. The main underlying goal in Local Property Testing is to distinguish objects that satisfy a given property from objects that are far from satisfying the property, using a small number of observations of the input object. Starting with the seminal works of [7, 13, 33], significant focus in the area has been devoted to locally testable error-correcting codes, called Locally Testable Codes (LTCs) [15]. LTCs are the key ingredients in several fundamental results in complexity theory, most notably in the PCP theorem [2, 3].

In this work we initiate the study of local testability for membership in *point lattices*, a class of infinite algebraic objects that form discrete subgroups of $\mathbb{R}^n$. Lattices are well-studied in mathematics, physics and computer science due to their rich algebraic structure [9]. Algorithms for various lattice problems have directly influenced the ability to solve integer programs [10, 23, 17]. Recently, lattices have found applications in modern cryptography due to attractive properties that enable efficient computations and security guarantees [28, 26, 31, 32]. Lattices are also used in practical communication settings to encode data in a redundant manner in order to protect it from channel noise during transmission [12].

A point lattice $L \subset \mathbb{R}^n$ of *rank* k and *dimension* n is specified by a set of linearly independent vectors $b_1, \ldots, b_k \in \mathbb{R}^n$ known as a basis, for some $k \leq n$. If $k = n$ the lattice is said to have full rank. The set L is defined to be the set of all vectors in $\mathbb{R}^n$ that are integer linear combinations of the basis vectors, i.e., $L := \{\sum_{i=1}^{k} \alpha_i b_i \mid \alpha_i \in \mathbb{Z} \; \forall \; i \in [k]\}$. Lattices are the analogues over $\mathbb{Z}$ of linear error-correcting codes over a finite field $\mathbb{F}$, which are generated as $\mathbb{F}$-linear combinations of a linearly independent set of basis vectors $b_1, \ldots, b_k \in \mathbb{F}^n$.

Given a basis for a lattice L, we are interested in testing if a given input $t \in \mathbb{R}^n$ belongs to L, or is far from all points in L by querying a small number of coordinates of t. We emphasize that this setting does not limit the computational space or time in pre-processing the lattice as well as the queried coordinates. The main goal is to design a tester that queries only a small number of coordinates of the input.

1.1 Motivation

Integer Programming. Lattices are the fundamental structures underlying integer programming problems. An integer programming problem (IP) is specified by a constraint matrix $A \in \mathbb{R}^{n \times m}$, a vector $b \in \mathbb{R}^n$. The goal is to verify if there exists an integer solution to the system $Ax = b, x \geq 0$. Although IP is NP-complete [18], its instances are solved routinely in practice using cutting planes and branch-and-cut techniques [35]. The relaxed problem of verifying integer feasibility of the system $Ax = b$ is equivalent to verifying whether b lies in the lattice generated by the columns of A. Thus, the relaxation problem is the membership testing problem in a lattice. It is solvable efficiently and is a natural pre-processing step to solving IPs. Furthermore, if the number of constraints n in the problem is very large, then it would be helpful to run a tester that reads only a partial set of coordinates of the input b to verify if b could lie in the lattice generated by the columns of A or is far from it. If the test rejects, then this saves on the computational effort to search for a non-negative solution.

Cryptography. In cryptographic applications, it is imperative to understand which lattices are difficult to test in order to ensure security of lattice-based cryptosystems. In some cryptanalytic attacks on lattice-based cryptosystems, one needs to distinguish target vectors that are close to lattice vectors from those that are far from all lattice vectors, a problem commonly known as the gap version of the Closest Vector Problem (GapCVP). An approach to address GapCVP is to use expensive distance estimation algorithms inspired by Aharonov and Regev [1] and Liu *et al.* [24]. Local testing of lattices is closely related to both distance estimation [30] and GapCVP, and hence progress in the proposed testing model could lead to new insights in cryptanalytic attacks.

Complexity theory. Lattices can be seen as coding theoretic objects naturally bringing features of error-correcting codes from the finite field domain to the real domain. As such,

a study of local testing (and correction) procedures for lattices naturally extends the classical notions of Locally Testable Codes (LTCs) and Locally Decodable Codes (LDCs), which are in turn of significance to computational complexity theory (for example in constructing probabilistically checkable proofs and hardness amplification, among numerous other applications). Characterizing local testability, explicitly initiated by Kaufman and Sudan [19], has been an intensely investigated direction in the study of LTCs. We believe that an analogous investigation of lattices is likely to bring new insights and new connections in property testing.

Lattice-based communication. Lattices are a major technical tool in communication systems as the analogue of error-correcting codes over reals, for applications such as wireless communication and transmission over analog lines. In lattice-coding, the message m is mapped to a point c in a chosen lattice L. The codeword c is transmitted over an analog channel. If the encoded message gets corrupted by the channel, then the channel output may not be a lattice point, thus enabling transmission error detection. In order to correct errors, computationally expensive decoding algorithms are employed. Instead, the receiver may perform a local test for membership in the lattice beforehand, allowing the costly decoding computation to run only when there is a reasonably high chance of correct decoding.

We now give an informal description of our testing model motivated by its application in lattice-coding. The transmission of each coordinate of a lattice-codeword over the analog channel consumes power that is proportional to the square of the transmitted value. Thus the power consumption for transmitting the lattice-codeword $c \in L \subset \mathbb{R}^n$ is proportional to its squared ℓ_2 norm. The power consumption for transmitting a codeword over the channel is usually constrained by a power budget. The noise vector is also subject to a bound on its power. The power budget for transmission is typically formulated by considering the lattice-code $C(L)$ defined by the set of lattice points $c \in L$ that satisfy $\sum_{i=1}^{n} c_i^2 \leq \sigma n$ for some constant power budget $\sigma > 0$. In order to ensure that the receiver can tolerate adversarial noise budget δ per channel use, the shortest nonzero vector $v \in L$ should be such that $\sum_{i=1}^{n} v_i^2 \geq \delta n$. Thus, the *relative distance* of the lattice-code $C(L)$ is defined to be $\sum_{i=1}^{n} v_i^2/n$, where $v \in L$ is a shortest nonzero lattice vector. The *rate* of a lattice-code $C(L)$ is defined to be $(1/n)\log|C(L)|$ (note that this quantity could be larger than 1). An *asymptotically good family of lattices*, in this work, is one that achieves rate and relative distance that are both lower bounded by a positive constant. Such families are ideal for use in noisy communication channels.

We define a notion of a tester that will be useful as a pre-processor for decoding, and is similar to the established notion of code testing: An ℓ_2-tester of a lattice L for a given distance parameter $\epsilon > 0$ is a probabilistic procedure that given an input $t \in \mathbb{R}^n$, queries at most q coordinates of t, accepts with probability at least $2/3$ if $t \in L$, and rejects with probability at least $2/3$ if $\sum_{i=1}^{n}(t_i - w_i)^2 \geq \epsilon n$ for every $w \in L$.

For the purposes of lattice-coding, the central lattice testing problem is whether there exists an asymptotically good family of lattices that can be tested for membership with query complexity $q = O(1)$.

1.2 Testing model

In the above application, we focused on ℓ_2 distances. We now formalize the notion of testing lattices for ℓ_p distances. We consider ℓ_p distances since these are natural notions for real-valued inputs [5]. The ℓ_p distance between $x, y \in \mathbb{R}^n$ is defined as $d_p(x, y) := \|x - y\|_p =$

$(\sum_{i\in[n]} |x_i - y_i|^p)^{1/p}$. The distance from $v \in \mathbb{R}^n$ to L is $d_p(v, L) := \min_{u\in L} d_p(v, u)$. Denote the ℓ_p norm of the real vector 1^n by $\|1^n\|_p$. For a lattice L, we denote the subspace of the lattice by $span(L)$. We focus on integral lattices, which are sub-lattices of $\mathbb{Z}^n$, as these are the most commonly encountered lattices in applications[1].

▶ **Definition 1** (Local test for lattices). An ℓ_p-tester $T(\epsilon, c, s, q)$ for a lattice $L \subseteq \mathbb{Z}^n$ is a probabilistic algorithm that queries q coordinates of the input $t \in \mathbb{R}^n$, and

- (completeness) *accepts* with probability at least $1 - c$ if $t \in L$,
- (soundness) *rejects* with probability at least $1 - s$ if $d_p(t, L) \geq \epsilon \cdot \|1^n\|_p$ (we call such a vector t to be ϵ-far from L).

If T always accepts inputs t that are in the lattice L then it is called 1-sided, otherwise it is 2-sided. If the queries performed by T depend on the answers to the previous queries, then T is called adaptive, otherwise it is called non-adaptive.

A test $T(\epsilon, 0, 0, q)$ is a test with perfect completeness and perfect soundness. 1-sided testers (i.e., testers with perfect completeness) are useful as a pre-processing step, as mentioned earlier. An asymptotically good family of lattices $L(n)$ for ℓ_p distances is one that has ℓ_p-relative distance lower bounded by a constant (i.e., $\min_{v\in L(n)} \|v\|_p^p/n = \Omega(1)$) and has $2^{\Omega(n)}$ lattice points in the origin-centered ℓ_p-ball of radius $n^{1/p}$. Similar to the application in lattice-coding and locally testable codes, a main question in ℓ_p-testing of lattices is the following:

▶ **Question 1.** *Is there an asymptotically good family of lattices that can be tested for membership with constant number of queries?*

Motivated by the applications in IP and cryptography, we identify another fundamental question in ℓ_p-testing of lattices:

▶ **Question 2.** *What properties of a given lattice enable the design of ℓ_p-testers with constant query complexity?*

Tolerant Testing. Many applications can tolerate a small amount of noise in the input. Parnas *et al.* [30] introduced the notion of tolerant testing to account for a small amount of noise in the input. Tolerant testing has been studied in the context of codes (e.g. [16, 20]), and in the context of properties of real-valued data in the ℓ_p norm (e.g. [5]). We extend the tolerant testing model to lattices as follows.

▶ **Definition 2** (Tolerant local test for lattices). An ℓ_p-tolerant-tester $T(\epsilon_1, \epsilon_2, c, s, q)$ for a lattice $L \subseteq \mathbb{Z}^n$ is a probabilistic algorithm that queries q coordinates of the input $t \in \mathbb{R}^n$, and

- (completeness) *accepts* with probability at least $1 - c$ if $d_p(t, L) \leq \epsilon_1 \cdot \|1^n\|_p$,
- (soundness) *rejects* with probability at least $1 - s$ if $d_p(t, L) \geq \epsilon_2 \cdot \|1^n\|_p$.

Tolerant testing with parameter $\epsilon_1 = 0$ corresponds to the notion of testing given in Definition 1. Tolerant testing and distance approximation are closely related notions. In fact, in the Hamming space, the ability to perform tolerant testing for *every* choice of $\epsilon_1 < \epsilon_2$ can be exploited to approximate distances (using a binary search) [30].

[1] Arbitrary lattices can be approximated by rational lattices and rational lattices can be scaled to integral lattices.

Analogy with code testers. A common notion of testing for membership in *error-correcting codes* requires that inputs at *Hamming distance* at least ϵn from the code be rejected. (This notion is only relevant when the covering radius of the code is larger than ϵn.) We include the common definition here, and note that stronger versions of testing have also been considered in the literature [15, 16].

▶ **Definition 3** (Local test for codes). A tester $T(\epsilon, c, s, q)$ for an error-correcting code $C \subseteq \mathbb{F}^n$ is a probabilistic algorithm that makes q queries to the input $t \in \mathbb{F}^n$, and

- (completeness) *accepts* with probability at least $1 - c$ if $t \in C$, and
- (soundness) *rejects* with probability at least $1 - s$ if $d_H(t, C) \geq \epsilon \cdot n$, where $d_H(u, v) := |\{i \in [n] : u(i) \neq v(i)\}|$ denotes the Hamming distance between u and v, and $d_H(t, C) := \min_{c \in C} d_H(t, c)$ (we call such a vector t to be ϵ-far from C).

1.3 Our contributions

We initiate the study of membership testing in point lattices from the perspective of sublinear algorithms aiming to lay the ground work for further advances towards resolving Question 1 and Question 2. Our contributions draw on connections between lattices and codes, and on well-known techniques in property testing.

1.3.1 Upper and lower bounds for testing specific lattice families

Motivated by applications in lattice-based communication, we focus on an asymptotically good family of sets constructed from linear codes, via the so-called "code formula" [12]. We show upper and lower bounds on the query complexity of ℓ_1-testers for code formulas, as a function of the query complexity of the constituent code testers.

Code formula lattices. For simplicity, in what follows we will slightly abuse notation and use binary code $C \subseteq \{0,1\}^n$ to denote both the code viewed over the field $\mathbb{F}_2 = \{0,1\}$ and the code embedded into $\mathbb{R}^n$ via the trivial embedding $0 \mapsto 0$ and $1 \mapsto 1$. All the arithmetic operations in the code formula refer to operations in $\mathbb{R}^n$. For two sets A and B of vectors we define $A + B := \{a + b \mid a \in A, b \in B\}$.

▶ **Definition 4** (Code Formula). Let $C_0 \subseteq C_1 \subseteq \cdots \subseteq C_{m-1} \subseteq C_m = \mathbb{F}_2^n$ be a family of nested binary linear codes. Then the code formula constructed from the family is defined as

$$C_0 + 2C_1 + \cdots + 2^{m-1}C_{m-1} + 2^m\mathbb{Z}^n.$$

Here, m is the *height* of the code-formula.

If the family satisfies the *Schur product condition*, namely, $c_1 * c_2 \in C_{i+1}$ for all codewords $c_1, c_2 \in C_i$, where the '*' operator is the coordinate-wise (Schur) product $c_1 * c_2 = \langle (c_1)_i \cdot (c_2)_i \rangle_{i \in [n]}$, then the code-formula forms a *lattice* (see [21]) and we denote it by $L(\langle C_i \rangle_{i=0}^{m-1})$.

Significance of code formula lattices. Code formula lattices with height one already have constant rate if the constituent code C_0 has minimum Hamming distance $\Omega(n)$. Unfortunately, these lattices have tiny relative minimum distance (since $2\mathbb{Z}^n$ has constant length vectors). However, code formulas of larger height achieve much better relative distance. In particular, it is easy to see that code formula lattices of height $m \geq \log n$ in which each of the constituent codes C_i has minimum Hamming distance $\Omega(n)$ give asymptotically good families of lattices [14, 9]. The code formula lattice constructed from a family of codes that satisfies the Schur-product condition is equivalent to the lattice constructed from the same

family of codes by Construction D [22, 9, 21]. Construction-D lattices are primarily used in communication settings, e.g. see Forney [12].

In this work we design a tester for code formula lattices using testers for the constituent codes.

▶ **Theorem 1.** *Let $0 < \epsilon, s < 1$ and $C_0 \subseteq C_1 \subseteq \cdots \subseteq C_{m-1} \subseteq \{0,1\}^n$ be a family of binary linear codes satisfying the Schur product condition. Suppose every C_i has a 1-sided tester $T_i(\epsilon/m2^{i+1}, 0, s, q_i)$. Then, there exists an ℓ_1-tester $T(\epsilon, 0, s, q)$ for the lattice $L(\langle C_i \rangle_{i=0}^{m-1})$ with query complexity*

$$q = O\left(\frac{1}{\epsilon}\log\frac{1}{s}\right) + \sum_{i=1}^{m-1} q_i.$$

Next, we show a lower bound on the query complexity for testing membership in code formula lattices, using lower bounds for testing membership in the constituent codes.

▶ **Theorem 2.** *Let $0 < \epsilon, c, s < 1$ and $C_0 \subseteq C_1 \subseteq \cdots \subseteq C_{m-1} \subseteq \{0,1\}^n$ be a family of binary linear codes satisfying the Schur product condition. Let $q_i = q_i(\epsilon, c, s)$ be such that any (possibly adaptive, 2-sided) ℓ_1-tester $T_i(\epsilon, c, s, q')$ for C_i satisfies $q' = \Omega(q_i)$, for every $i = 0, 1, \ldots, m-1$. Then every (possibly adaptive, 2-sided) ℓ_1-tester $T(\epsilon, c, s, q)$ for the lattice $L(\langle C_i \rangle_{i=0}^{m-1})$ has query complexity*

$$q = \Omega\left(\max\left\{\frac{1}{\epsilon}\log\frac{1}{s}, \max_{i=0,1,\ldots,m-1} q_i\right\}\right).$$

Code formula lattices from Reed-Muller codes. We instantiate the upper and lower bounds on the query complexity for a common family of code formula lattices constructed using Reed-Muller codes [12] to obtain nearly matching upper and lower bounds. We recall Reed-Muller codes below.

▶ **Definition 5** (Reed Muller Codes)**.** Each codeword of a binary Reed-Muller code $RM(k, r) \subseteq \mathbb{F}_2^{2^r}$ corresponds to a polynomial $p(x) \in \mathbb{F}_2[x]$ in r variables of degree at most k evaluated at all 2^r possible inputs $x \in \mathbb{F}_2^r$.

For the family of Reed-Muller codes in $\mathbb{F}_2^{2^r}$, it is well-known that $RM(0, r) \subseteq RM(1, r) \subseteq RM(2, r) \subseteq RM(3, r) \subseteq \cdots \subseteq RM(r-1, r) \subseteq RM(r, r) = \mathbb{F}_2^{2^r}$. A particular family of RM codes that leads to code formula lattices is $\langle RM(k_i, r) \rangle_{i=0}^{\log r}$, with $k_i = 2^i$. Indeed, it can be easily verified that this family satisfies the Schur product condition since Reed-Muller codewords are evaluation tables of multivariate polynomials over the binary field and product of two degree k polynomials is a degree $2k$ polynomial. Hence for height $m \leq \log r$ the construction $\langle RM(2^i, r) \rangle_{i=0}^{m-1}$ gives rise to a lattice. We note these lattices have small relative minimum distance and are not asymptotically good families of lattices.

▶ **Corollary 3.** *Let $0 \leq k_0 < k_1 < \cdots < k_{m-1} < r$ be integers such that the family of Reed-Muller codes $RM(k_0, r) \subseteq RM(k_1, r) \subseteq \cdots \subseteq RM(k_{m-1}, r)$ satisfies the Schur product condition. Let $0 < \epsilon, s < 1$ and L be the lattice obtained from this family of codes using the code formula construction:*

$$L = RM(k_0, r) + 2RM(k_1, r) + \cdots + 2^{m-1}RM(k_{m-1}, r) + 2^m\mathbb{Z}^{2^r}.$$

Then, there exists an ℓ_1-tester $T(\epsilon, 0, s, q)$ for L with query complexity

$$q(\epsilon, s) = O\left(2^{k_{m-1}} \cdot \frac{1}{\epsilon}\log\frac{1}{s}\right).$$

In particular, when the height m and the degrees are constant, the query complexity of the tester is a constant.

For the lower bound, we obtain the following corollary using known lower bounds for testing Reed-Muller codes.

▶ **Corollary 4.** *Let $0 \leq k_0 < k_1 < \cdots < k_{m-1} < r$ be integers such that the family of Reed-Muller codes $RM(k_0, r) \subseteq RM(k_1, r) \subseteq \cdots \subseteq RM(k_{m-1}, r)$ satisfies the Schur product condition. Let $0 < \epsilon, c, s < 1$ be constants and L be the lattice obtained from this family of codes using the code formula construction:*

$$L = RM(k_0, r) + 2RM(k_1, r) + \cdots + 2^{m-1}RM(k_{m-1}, r) + 2^m\mathbb{Z}^{2^r}.$$

Then, every (possibly 2-sided, adaptive) ℓ_1-tester $T(\epsilon, c, s, q)$ for L has query complexity

$$q = \Omega(2^{k_{m-1}}).$$

We note that for code formula lattices obtained from Reed-Muller codes, Corollaries 3 and 4 show matching bounds (up to a constant factor depending on ϵ, s).

Random lattices. There exists a distribution of random lattices which are impossible to test with small number of queries. This follows from Theorem 2 and considering random codes, which typically need at least a linear number of queries to test. We illustrate a concrete example by considering the following distribution of random lattices [11, 4]: For constants $b < a$, let $m = nb/a$ and let $H \in \mathbb{F}_2^{m\times n}$ be a random matrix such that each row and column has exactly a and b non-zeroes respectively. Consider the linear code $C_{a,b} := \{x \in \mathbb{F}_2^n : Hx = 0 (\mathrm{mod}\ 2)\}$ and the code formula lattice $L(C_{a,b})$ associated with the linear code $C_{a,b}$.

▶ **Theorem 5.** *There exist constants a, b, ϵ, c, s such that every (possibly 2-sided, adaptive) ℓ_1-tester $T(\epsilon, c, s, q)$ for $L(C_{a,b})$ has query complexity $q = \Omega(n)$.*

The above theorem follows as an immediate corollary of Theorem 2 and Theorem 3.7 of [4].

1.3.2 Tolerant testing code formulas

We also obtain upper bounds for tolerantly testing code formula lattices.

▶ **Theorem 6.** *Let $0 < \epsilon_1, \epsilon_2, c, s < 1$ and $C_0 \subseteq C_1 \subseteq \cdots \subseteq C_{m-1} \subseteq \{0,1\}^n$ be a family of binary linear codes satisfying the Schur product condition. Suppose every C_i has a tolerant tester $T_i(2\epsilon_1, \frac{\epsilon_2}{m2^{i+1}}, \frac{c}{m+1}, s, q_i)$. Let $\gamma = \min\{c/(m+1), s\}$, $\epsilon_2 > m2^{m+1}\epsilon_1$. Then there exists an ℓ_1-tolerant-tester $T(\epsilon_1, \epsilon_2, c, s, q)$ for the lattice $L(\langle C_i\rangle_{i=0}^{m-1})$ with query complexity*

$$q = O\left(\frac{1}{(\epsilon_2 - 2\epsilon_1)^2}\log\left(\frac{1}{\gamma}\right)\right) + \sum_{i=0}^{m-1} q_i.$$

▶ **Corollary 7.** *Let $0 \leq k_0 < k_1 < \cdots < k_{m-1} < r$ be integers such that the family of Reed-Muller codes $RM(k_0, r) \subseteq RM(k_1, r) \subseteq \cdots \subseteq RM(k_{m-1}, r)$ satisfies the Schur product condition. Let L be the lattice obtained from this family of codes using the code formula construction:*

$$L = RM(k_0, r) + 2RM(k_1, r) + \cdots + 2^{m-1}RM(k_{m-1}, r) + 2^m\mathbb{Z}^{2^r}.$$

Then there exists a ℓ_1-tolerant-tester $T(\epsilon_1, \epsilon_2, 1/3, 1/3, q)$ for L for all $\epsilon_1 \leq \frac{c_1'}{2^{k_{m-1}}}$, $\epsilon_2 \geq \frac{c_2' m}{2^{k_0-1}}$ (for some constants c_1' and c_2') with query complexity $q = O(2^{k_{m-1}} \cdot \log m)$.

1.3.3 A canonical/linear test for lattices

We show a reduction from any given arbitrary test to a *canonical linear test*, thus suggesting that it is sufficient to design *canonical linear tests* for achieving low query complexity. In order to describe the intuition behind a canonical linear test, we first illustrate how to solve the membership testing problem when all coordinates of the input are known. For a given lattice L, its *dual lattice* is defined as

$$L^{\perp} := \{u \in span(L) \mid \langle u, v\rangle \in \mathbb{Z}, \text{ for all } v \in L\}.$$

It is easy to verify that $(L^{\perp})^{\perp} = L$. Furthermore, a vector $v \in L$ if and only if for all $u \in L^{\perp}$, we have $\langle u, v\rangle \in \mathbb{Z}$. Thus, to test membership of t in L in the classical decision sense, it is sufficient to verify whether t has integer inner products with a set of basis vectors of the dual lattice $L^{\perp}$. Inspired by this observation, we define a canonical *linear test* for lattices as follows. For a lattice $L \subseteq \mathbb{R}^n$ and $J \subseteq [n]$, let $L^{\perp}_J := \{x \in L^{\perp} \mid supp(x) \subseteq J\}$, where $supp(x)$ is the set of non-zero indices of the vector x.

▶ **Definition 6** (Linear Tester)**.** A *linear tester* for a lattice $L \subseteq \mathbb{Z}^n$ is a probabilistic algorithm which queries a subset $J = \{j_1, \ldots, j_q\} \subseteq [n]$ of coordinates of the input $t \in \mathbb{R}^n$ and accepts t if and only if $\langle t, x\rangle \in \mathbb{Z}$ for all $x \in L^{\perp}_J$. [2]

Remark. By definition, the probabilistic choices of a linear tester are only over the set of coordinates to be queried: upon fixing the coordinate queries, the choice of the algorithm to accept or reject is fully determined. Furthermore, a linear tester is 1-sided since if the input t is a lattice vector, then for every dual vector $u \in L^{\perp}$, the inner product $\langle u, t\rangle$ is integral, and so it will be accepted with probability 1.

We show that non-adaptive linear tests are nearly as powerful as 2-sided adaptive tests for a full-rank lattice. We reduce any (possibly 2-sided, and adaptive) test for a full-rank lattice to a non-adaptive linear test for the same distance parameter ϵ, with a small increase in the query complexity and the soundness error.

▶ **Theorem 8.** *Let $L \subseteq \mathbb{Z}^n$ be a lattice with $rank(L) = n$. If there exists an adaptive 2-sided ℓ_p-tester $T(\epsilon, c, s, q)$ with query complexity $q = q_T(\epsilon, c, s)$, then there exists a non-adaptive linear ℓ_p-tester $T'(\epsilon, 0, c+s, q')$ with query complexity $q' = q_T(\epsilon/2, c, s) + O((1/\epsilon^p)\log(1/s))$.*

Furthermore, if we are guaranteed that the inputs are in $\mathbb{Z}^n$, then the query complexity of the test T' above can be improved to be identical to that of T (up to a constant factor in the ϵ parameter). The increase in the query complexity comes from an extra step used to verify the integrality of the input.

Theorem 8 suggests that, for the purposes of designing a tester with small query complexity, it is sufficient to design a non-adaptive linear tester, i.e., it suffices to only identify the probability distribution for the coordinates that are queried. Moreover, this theorem makes progress towards Question 2, since it shows that a lower bound on the query complexity of non-adaptive linear tests for a particular lattice implies a lower bound on the query complexity of all tests for that lattice. Thus in order to understand the existence of low query complexity tester for a particular lattice, it is sufficient to examine the existence of low query complexity *non-adaptive linear* tester for that lattice.

---

[2] Verifying whether $\langle t, x\rangle \in \mathbb{Z}$ for all $x \in L^{\perp}_J$ can be performed efficiently by checking inner products with a set of basis vectors of the lattice $L^{\perp}_J$.

We note that Theorem 8 is the analogue of the result of [4] for linear error-correcting codes. In section 2, we comment on the comparison between our proof and that in [4].

1.3.4 Testing membership of inputs outside the span of the lattice

We also observe a stark difference between the membership testing problem for a linear code, and the membership testing problem for a lattice. In the membership testing problem for a linear code $C \subseteq \mathbb{F}^n$ defined over a finite field that is specified by a basis, the input is assumed to be a vector in $\mathbb{F}^n$ and the goal is to verify whether the input lies in the span of the basis (see definition 3). As opposed to codes, for a lattice $L \subseteq \mathbb{R}^n$, the input is an arbitrary real vector, and the goal is to verify whether the input is a member of L, and not to verify whether the input is a member of the span of the lattice. Thus, the inputs to the lattice membership testing problem could lie either in $span(L)$, or outside $span(L)$. Interestingly, for some lattices it is easy to show strong lower bounds on the query complexity if the inputs are allowed to lie outside $span(L)$, thus suggesting that such inputs are hard to test.

▶ **Theorem 9.** *Let $L \subseteq \mathbb{Z}^n$ be a lattice of rank k. Let $P \subseteq [n]$ be the support of the vectors in $span(L)^\perp$. Let $0 < \epsilon, c, s < 1$. Every non-adaptive ℓ_p-tester $T(\epsilon, c, s, q)$ for L for inputs in $\mathbb{R}^n$ has query complexity*

$$q = \Omega(|P|).$$

On the other hand, testers for inputs in the $span(L)$ can be lifted to obtain testers for all inputs (including inputs that could possibly lie outside $span(L)$).

▶ **Theorem 10.** *Let $L \subseteq \mathbb{Z}^n$ be a lattice of rank k. Let $P \subseteq [n]$ be the support of the vectors in $span(L)^\perp$. Let $0 < \epsilon, c, s < 1$, and suppose L has an ℓ_p-tester $T(\epsilon, c, s, q)$ for inputs $t \in span(L)$. Then L has a tester $T'(2\epsilon, c, s, q')$ for inputs in $\mathbb{R}^n$ with query complexity*

$$q' \leq q + |P|.$$

Theorem 10 implies that for lattices L of rank at most $n-1$, if the membership testing problem for inputs that lie in $span(L)$ is solvable using a small number of queries and if $span(L)^\perp$ is supported on few coordinates, then the membership testing problem for all inputs (including those that do not lie in $span(L)$) is solvable using a small number of queries.

Knapsack Lattices. Theorem 9 implies a linear lower bound for non-adaptively testing a well-known family of lattices, known as *knapsack lattices*, which have been investigated in the quest towards lattice-based cryptosystems [25, 34, 29]. We recall that a knapsack lattice is generated by a set of basis vectors $B = \{b_1, \ldots, b_{n-1}\}, b_i \in \mathbb{R}^n$ that are of the form

$$\begin{aligned} b_1 &= (1, 0, \ldots, 0, a_1) \\ b_2 &= (0, 1, \ldots, 0, a_2) \\ &\vdots \\ b_{n-1} &= (0, 0, \ldots, 1, a_{n-1}) \end{aligned}$$

where $a_1, \ldots, a_n$ are integers. We denote such a knapsack lattice by $L_{a_1,\ldots,a_{n-1}}$.

▶ **Corollary 11.** *Let $a_1, \ldots, a_n$ be integers and $0 < \epsilon, c, s < 1$. Every non-adaptive ℓ_p-tester $T(\epsilon, c, s, q)$ for $L_{a_1,\ldots,a_n}$ has query complexity*

$$q = \Omega(n).$$

However, knapsack lattices with bounded coefficients are testable with a constant number of queries if the inputs are promised to lie in $span(L)$.

▶ **Theorem 12.** *Let $a_1, \ldots, a_n$ be integers with $M = \max_{i\in[n]} |a_i|^p$ and $0 < \epsilon, s < 1$. There exists a non-adaptive ℓ_p-tester $T(\epsilon, 0, s, q)$ for $L_{a_1,\ldots,a_n}$ with query complexity $q = O\left(\frac{M}{\epsilon^p} \cdot \log \frac{1}{s}\right)$, if the inputs are guaranteed to lie in $span(L)$.*

Theorem 12 indicates that the large lower bound suggested by Theorem 9 could be circumvented for certain lattices if we are promised that the inputs lie in $span(L)$. The assumption that the input lies in $span(L)$ is natural in decoding problems for lattices.

2 Overview of the proofs

2.1 Upper and lower bounds for testing general code formula lattices

The constructions of a tester for Theorem 1 and a tolerant tester for Theorem 6 follow the natural intuition that in order to test the lattice one can test the underlying codes individually. The proof relies on a triangle inequality that can be derived for such lattices. The application to code-formula lattices constructed from Reed-Muller codes follows from the tight analysis of Reed-Muller code testing from [6], which guarantees constant rejection probability of inputs that are at distance proportional to the minimum distance of the code. We note that the time complexity of the code-formula tester is given by the sum of the run-times of the component code testers. Since the component code testers can be assumed to be linear, and hence efficient, the code-formula lattice tester is also efficient.

While the tester that we construct from code testers for the purposes of proving Theorem 1 is an adaptive linear test, there is a simple variant that is a non-adaptive linear test with at least as good correctness and soundness. (see Remark 5.16 in full version [8] for a formal description).

The lower bound (Theorem 2) relies on the fact that if an input t is far from the code C_k in the code formula construction, then the vector $2^k t$ is far from the lattice . Moreover, if $t \in C_k$ then $2^k t$ belongs to the lattice. Therefore a test for the lattice can be turned into a test for the constituent codes.

2.2 From general tests to canonical tests

We briefly outline our reduction for Theorem 8. Suppose $T(\epsilon, c, s, q)$ is a 2-sided, adaptive tester with query complexity $q = q_T(\epsilon, c, s)$ for a full rank integral lattice L. Such a tester handles all real-valued inputs. We first restrict T to a test that processes only integral inputs in the bounded set $\mathcal{Z}_d = \{0, 1, \ldots, d-1\}$ (for some carefully chosen d), and so the restricted test inherits all the parameters of T. We remark that $\mathcal{Z}_d \subset \mathbb{Z}$ is a subset of integers, and it should not be confused with $\mathbb{Z}_d$, the ring of integers modulo d.

A key ingredient in our reduction is choosing the appropriate value of d in order to enable the same guarantees as that of codes. We choose d such that $d\mathbb{Z}^n \subseteq L$. Such a d always exists [27]. This choice of d allows us to add any vector in $V = L \bmod d$ (embedded in $\mathbb{R}^n$) to any vector $x \in \mathbb{R}^n$ without changing the distance of x to L in any ℓ_p-norm (see Proposition 14).

Since our inputs are now integral and bounded, any adaptive test can be viewed as a distribution over deterministic tests, which themselves can be viewed as decision trees. This

allows us to proceed along the same lines as in the reduction for codes over finite fields of [4].

We exploit the property that adding any vector in V to any vector $x \in \mathbb{R}^n$ does not change the distance to L. In the first step of our reduction we add a random vector in V to the input and perform a probabilistic *linear* test. The idea is that one can relabel the decision tree of any test according to the decision tree of a linear test, such that the error shifts from the positive (yes) instances to the negative (no) instances (see Lemma 15). A simple property of lattices used in this reduction is that if the set of queries I and answers a_I do not have a local witness for non-membership in the lattice (in the form of a dual lattice vector v supported on I such that $\langle w_I, v_I \rangle \notin \mathbb{Z}$), then there exists $w \in L$ that extends a_I to the remaining set of coordinates (i.e., $a_I = w_I$).

In the next step we remove the adaptive aspect of the test to obtain a non-adaptive linear test for inputs in $\mathcal{Z}_d^n$ (see Lemma 16). We obtain this tester by performing the adaptive queries on a randomly chosen vector in V (and not on the input itself) and rejecting/accepting according to whether there exists a local witness for the non-membership of the input queried on the same coordinates.

We then lift this test to a non-adaptive linear test for inputs in $\mathbb{Z}^n$, by simulating the test over $\mathcal{Z}_d^n$ on the same queried coordinates but using the answers obtained after taking modulo d. Owing to the choice of d, this does not change the distance of the input to the lattice (see Lemma 17).

Finally, we extend this test to a non-adaptive linear test for inputs in $\mathbb{R}^n$ by performing some additional queries to rule out inputs that are not in $\mathbb{Z}^n$. For this, we design a tester for the integer lattice $\mathbb{Z}^n$ with query complexity $O((1/\epsilon^p)\log(1/s))$. This final step of testing integrality increases the overall query complexity to $q_T(\epsilon/2, c, s) + O((1/\epsilon^p)\log(1/s))$ (see Lemma 18).

Organization. We present the formal lemmas needed to prove Theorem 8 in Section 3. We refer the reader to the full version [8] for all the missing proofs.

3 Reducing an arbitrary test to a non-adaptive linear test

In this section we sketch the proof of Theorem 8. Throughout this section, we focus on full-rank integral lattices. Given a 2-sided adaptive ℓ_p-tester $T(\epsilon, c, s, q)$, with $q = q_T(\epsilon, c, s)$ for an integral lattice L, we construct a non-adaptive linear ℓ_p-tester $T'(\epsilon, 0, c+s, q)$ with query complexity $q' = q_T(\epsilon/2, c, s) + O((1/\epsilon^p)\log(1/s))$. We reduce the inputs to a bounded set using the following property of integral lattices.

▶ **Fact 13.** *[27] Given any full rank integral lattice L, there exists $d \in \mathbb{Z}$ such that $d \cdot \mathbb{Z}^n \subseteq L$. In particular $|\det(L)| \cdot \mathbb{Z}^n \subseteq L$ for any lattice (where $\det(L)$ denotes the determinant of a lattice, a parameter that can be computed given a basis of the lattice). For instance, we can take $d = 2^m$ for the lattices of height m obtained using the code formula construction.*

Let $V = L \bmod d$ embedded in $\mathbb{Z}^n$ (i.e., we treat V as a set of vectors in $\mathbb{Z}^n$ each of which is obtained by taking coordinate-wise modulo d of some lattice vector). Thus, $V \subseteq \mathcal{Z}_d^n$. We will need the following properties of V.

▶ **Proposition 14.** *Let $L \subseteq \mathbb{Z}^n$ be a full-rank lattice, $d \in \mathbb{Z}_+$ such that $d\mathbb{Z}^n \subseteq L$, and let $V = L \bmod d \subseteq \mathbb{Z}^n$. Then V satisfies the following properties:*

1. *$v \in L$ if and only if $v \bmod d \in V$.*
2. *$V = L \cap \mathcal{Z}_d^n$.*

3. $(v + V) \bmod d \subseteq V$ *if and only if* $v \in L$.
4. *For any* $v \in \mathbb{Z}^n$, $d_p(v, L) = d_p(v \bmod d, L)$.

Theorem 8 will immediately follow by combining Lemmas 15, 16, 17, and 18.

▶ **Lemma 15.** *Suppose a full-rank lattice* $L \subseteq \mathbb{Z}^n$ *with* $d\mathbb{Z}^n \subseteq L$ *for* $d \in \mathbb{Z}_+$ *has an adaptive 2-sided* ℓ_p*-tester* $T(\epsilon, c, s, q)$ *for inputs from the domain* $\mathcal{Z}_d^n$. *Then* L *has an adaptive linear* ℓ_p*-tester* $T'(\epsilon, 0, c + s, q)$ *for inputs from the domain* $\mathcal{Z}_d^n$.

▶ **Lemma 16.** *Suppose a full-rank lattice* $L \subseteq \mathbb{Z}^n$ *with* $d\mathbb{Z}^n \subseteq L$ *for* $d \in \mathbb{Z}_+$ *has an adaptive linear* ℓ_p*-tester* $T(\epsilon, 0, s, q)$ *for inputs from the domain* $\mathcal{Z}_d^n$. *Then* L *has a non-adaptive linear* ℓ_p*-tester* $T'(\epsilon, 0, s, q)$ *for inputs from the domain* $\mathcal{Z}_d^n$.

▶ **Lemma 17.** *Let* $L \subseteq \mathbb{Z}^n$ *be a full-rank lattice with* $d\mathbb{Z}^n \subseteq L$ *for* $d \in \mathbb{Z}_+$. *Then,* L *has a non-adaptive linear* ℓ_p*-tester* $T(\epsilon, 0, s, q)$ *for inputs from the domain* $\mathcal{Z}_d^n$ *if and only if* L *has a non-adaptive linear* ℓ_p*-tester* $T'(\epsilon, 0, s, q)$ *for inputs from the domain* $\mathbb{Z}^n$.

▶ **Lemma 18.** *Suppose a full-rank lattice* $L \subseteq \mathbb{Z}^n$ *has a non-adaptive* ℓ_p*-tester* $T(\epsilon, c, s, q)$ *for inputs from the domain* $\mathbb{Z}^n$. *Then there exists a non-adaptive* ℓ_p*-tester* $T'(\epsilon, c, s, q')$ *for inputs in* $\mathbb{R}^n$ *with query complexity* $q' = q(\epsilon/2, c, s) + O((1/\epsilon^p) \log(1/s))$. *Moreover, if* T *is a linear tester, then so is* T'.

The proof of Lemma 18 uses the following tester for integer lattices which is based on querying a random collection of coordinates and verifying whether all of them are integral.

▶ **Lemma 19.** *For every* $0 < \epsilon \leq 1$ *and every* $0 < s \leq 1$, *there exists a non-adaptive linear* ℓ_p*-tester* $T_p(\epsilon, 0, s, q_Z)$ *for* $\mathbb{Z}^n$ *with query complexity*

$$q_Z = O\left(\frac{1}{\epsilon^p} \log \frac{1}{s}\right).$$

4 Discussion

In this paper we defined a notion of local testing for a new family of objects: point lattices. Our results demonstrate connections between lattice testing and the ripe theory of locally testable codes, and brings up numerous avenues for further research (particularly, Questions 1 and 2).

We remark that the notion of being 'ϵ-far' from the lattice may be defined differently than in Definition 1, depending on the application of interest. In particular, in applications like IP and cryptography, it is natural to ask for a notion of tester that ensures that scaling the lattice does not change the query complexity. An alternate definition of ϵ-far based on the *covering radius* of the lattice could be helpful to achieve this property. The covering radius of a lattice $L \subseteq \mathbb{R}^n$ (similar to codes) is the largest distance of any vector in $\mathbb{R}^n$ to the lattice. It is trivial to design a tester to verify if a point is in the lattice or at distance more than the covering radius from the lattice (simply accept all inputs). In order to have a tester notion where scaling preserves query complexity, we may define a vector as being ϵ-far from the lattice, if the distance of the vector to every lattice point is at least ϵ times the covering radius of the lattice. We note that the covering radius of any *integral lattice* is $\Omega(\|1^n\|_p)$. Indeed, the densest possible integral lattice, namely the integer lattice $\mathbb{Z}^n$, has covering radius $(1/2)\|1^n\|_p$, as exhibited by the point $v = (1/2, \ldots, 1/2) \in \mathbb{R}^n$. Thus, by asking the tester to reject points at distance more than $\epsilon\|1^n\|_p$ in Definition 1, we have

settled upon a strong notion of being ϵ-far from the lattice (i.e., the definition would in particular imply that vectors that are farther than ϵ times the covering radius would be rejected by the tester). This definition is essentially equivalent to the current Definition 1 if the covering radius of the lattice is $\Theta(n)$. With the modified definition of local testers using covering radius as described above, the equivalent Question 1 is to identify a family of lattices that can be tested using a constant number of queries, achieves constant rate and whose ratio of minimum distance to covering radius is also at least a constant.

Acknowledgments. We thank Chris Peikert for mentioning to us about the potential application to cryptanalysis, and anonymous reviewers for helpful comments and pointers.

References

1 Dorit Aharonov and Oded Regev. Lattice problems in $NP \cap coNP$. *J. ACM*, 52(5):749–765, 2005.

2 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998.

3 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998.

4 E. Ben-Sasson, P. Harsha, and S. Raskhodnikova. Some 3CNF properties are hard to test. *SIAM Journal on Computing*, 35(1):1–21, 2005. Earlier version in STOC'03.

5 Piotr Berman, Sofya Raskhodnikova, and Grigory Yaroslavtsev. L_p-testing. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 164–173, 2014.

6 Arnab Bhattacharyya, Swastik Kopparty, Grant Schoenebeck, Madhu Sudan, and David Zuckerman. Optimal testing of Reed-Muller codes. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 488–497, 2010.

7 M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47:549–595, 1993.

8 Karthekeyan Chandrasekaran, Mahdi Cheraghchi, Venkata Gandikota, and Elena Grigorescu. Local testing for membership in lattices. *arXiv preprint arXiv:1608.00180*, 2016.

9 J. Conway, N.J.A. Sloane, and E. Bannai. *Sphere Packings, Lattices and Groups*. A series of comprehensive studies in mathematics. Springer, 1999.

10 Friedrich Eisenbrand. Fast integer programming in fixed dimension. In *Algorithms - ESA 2003, 11th Annual European Symposium, Budapest, Hungary, September 16-19, 2003, Proceedings*, pages 196–207, 2003.

11 Uri Erez, Simon Litsyn, and Ram Zamir. Lattices which are good for (almost) everything. *IEEE Transactions on Information Theory*, 51(10):3401–3416, 2005.

12 G. D. Forney. Coset codes-I: Introduction and geometrical classification. *IEEE Transactions on Information Theory*, 34(5):1123–1151, 1988.

13 K. Friedl and M. Sudan. Some improvements to low-degree tests. In *Proceedings of the 3rd Annual Israel Symposium on Theory and Computing Systems*, 1995.

14 Philippe Gaborit and Gilles Zémor. On the construction of dense lattices with a given automorphisms group. In *Annales de l'institut Fourier*, volume 57, pages 1051–1062, 2007.

15 Oded Goldreich. Short locally testable codes and proofs: A survey in two parts. In *Property Testing - Current Research and Surveys*, pages 65–104, 2010.

16 Venkatesan Guruswami and Atri Rudra. Tolerant locally testable codes. In *Proceedings of RANDOM/APPROX 2005*, pages 306–317, 2005.

17 Ravi Kannan. Minkowski's convex body theorem and integer programming. *Math. Oper. Res.*, 12(3):415–440, August 1987.

18 Richard M. Karp. Reducibility among combinatorial problems. In *Proceedings of a symposium on the Complexity of Computer Computations*, pages 85–103, 1972.

19 T. Kaufman and M. Sudan. Algebraic property testing: The role of invariance. In *STOC*, pages 403–412, 2008.

20 Swastik Kopparty and Shubhangi Saraf. Tolerant linearity testing and locally testable codes. In *Proceedings of RANDOM*, pages 601–614, 2009.

21 Wittawat Kositwattanarerk and Frédérique E. Oggier. Connections between construction D and related constructions of lattices. *Des. Codes Cryptography*, 73(2):441–455, 2014.

22 John Leech and NJA Sloane. Sphere packings and error-correcting codes. *Canad. J. Math*, 23(4):718–745, 1971.

23 H.W Lenstra Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.

24 Yi-Kai Liu, Vadim Lyubashevsky, and Daniele Micciancio. On bounded distance decoding for general lattices. In *Proceedings of RANDOM*, pages 450–461, 2006.

25 Ralph C Merkle and Martin E Hellman. Hiding information and signatures in trapdoor knapsacks. *IEEE Transactions on Information Theory*, 24(5):525–530, 1978.

26 Daniele Micciancio. *The LLL Algorithm: Survey and Applications*, chapter Cryptographic functions from worst-case complexity assumptions, pages 427—-452. Information Security and Cryptography. Springer, December 2009. Prelim. version in Proc. of LLL25, 2007.

27 Daniele Micciancio. Lecture notes on lattice algorithms and applications, Winter 2012, Lecture 2, 2012.

28 Daniele Micciancio and Shafi Goldwasser. *Complexity of Lattice Problems: a cryptographic perspective*, volume 671 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, Massachusetts, March 2002.

29 Andrew M Odlyzko. The rise and fall of knapsack cryptosystems. *Cryptology and computational number theory*, 42:75—-88, 1990.

30 M. Parnas, D. Ron, and R. Rubinfeld. Tolerant property testing and distance approximation. *Journal of Computer and System Sciences*, 72(6):1012–1042, 2006.

31 Oded Regev. Lattice-based cryptography. In *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, pages 131–141, 2006.

32 Oded Regev. The learning with errors problem (invited survey). In *IEEE Conference on Computational Complexity*, pages 191–204, 2010.

33 R. Rubinfeld and M. Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25:252–271, 1996.

34 Adi Shamir. A polynomial time algorithm for breaking the basic merkle-hellman cryptosystem. In *Advances in Cryptology*, pages 279–288. Springer, 1983.

35 Laurence A Wolsey and George L Nemhauser. *Integer and combinatorial optimization*. John Wiley & Sons, 2014.