The Geometry of Lattice Cryptography

Daniele Micciancio*

February 16, 2012

Abstract

Lattice cryptography is one of the hottest and fastest moving areas in mathematical cryptography today. Interest in lattice cryptography is due to several concurring factors. On the theoretical side, lattice cryptography is supported by strong worst-case/average-case security guarantees. On the practical side, lattice cryptography has been shown to be very versatile, leading to an unprecedented variety of applications, from simple (and efficient) hash functions, to complex and powerful public key cryptographic primitives, culminating with the celebrated recent development of fully homomorphic encryption. Still, one important feature of lattice cryptography is simplicity: most cryptographic operations can be implemented using basic arithmetic on small numbers, and many cryptographic constructions hide an intuitive and appealing geometric interpretation in terms of point lattices. So, unlike other areas of mathematical cryptology even a novice can acquire, with modest effort, a good understanding of not only the potential applications, but also the underlying mathematics of lattice cryptography.

In these notes, we give an introduction to the mathematical theory of lattices, describe the main tools and techniques used in lattice cryptography, and present an overview of the wide range of cryptographic applications. This material should be accessible to anybody with a minimal background in linear algebra and some familiarity with the computational framework of modern cryptography, but no prior knowledge about point lattices.

1 Introduction

Lattice cryptography is one of the most fascinating areas of mathematical cryptography. Supported by a solid theoretical foundation, lattice cryptography can also be very attractive in practice, as an alternative to more traditional solutions based on number theory. One of the features that makes lattice cryptography potentially attractive in practice is its simplicity: the most fundamental operation of lattice cryptography is just a modular integer matrix-vector multiplication $\mathbf{Ax} \mod q$. Typically, q is a small integer, which comfortably fits in a machine level register, so that all the basic operations can be efficiently performed (and easily implemented) without the need of a "big-num" library for arbitrary precision arithmetic. While many cryptographic functions based on lattices can be easily described using matrix notation (and this is how they are typically described in most research papers), the operations they perform and the computational problems they hide are fundamentally geometric.

The matrix formulation may be all that is needed to specify and implement lattice cryptography. But in order to truly understand it and develop an intuition of what makes lattice cryptography secure, one needs to look under the hood, and explore the geometric structure of the underlying lattices. As we will see, while quite different from the mathematics used in more conventional cryptography, the mathematics of lattices (or at least most of the mathematics of lattices used in cryptography) is not that hard to understand. In fact, once you learn the most basic concepts, it is pretty simple! After all, geometry is one of the most intuitive areas of mathematics, well grounded in our everyday experiences in the physical world, and many of the

*An edited version of this paper appears in Springer LNCS volume 6858, *Foundations of Security Analysis and Design VI - FOSAD Tutorial Lectures, pp. 185-210.* Supported in part by the National Science Foundation under Grant CNS-0831536. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

concepts we will introduce can be easily illustrated by pictures.[1] If you can draw a picture that captures the intuition of why factoring the product of two large prime numbers is a computationally hard problem, I would like to see your picture! So, let us leave our fears behind, and move on with lattices.

In these notes, we will first learn about lattices, their geometric structure, and the computational problems they define. Then we will look at how these simple geometric object can be used for the construction of cryptographic functions. Beside the potential for efficient implementation, lattice cryptography is very attractive because of its versatility: using lattices, in the last few years, researchers were able to develop solutions for an incredibly rich set of security problems, from simple (but efficient) hash functions [57, 68, 48, 51], to hierarchical identity based public key encryption [34, 22, 1, 2], and much more. As of this writing, the latest and greatest discovery in lattice cryptography is the development of fully homomorphic encryption, pioneered by Gentry in [28], and still a very fast moving research target [82, 29, 65, 79, 80, 32, 25, 19, 31, 30]. In these notes, you will not learn about the most complex applications of lattices, including fully homomorphic encryption, but you will learn enough about lattices and lattice cryptography to proceed on your own and read research papers in the area. Beside an introduction to the fundamentals of lattice cryptography, this paper includes a extensive bibliography, that can be used as a source of pointers for further study.

Notational conventions. We use $\mathbb{Z}$ for the set of integers, and $\mathbb{R}$ for the set of real numbers. Elements of these sets are denoted by lowercase letters. Sets are denoted with uppercase letters. We use bold lower case letters $\mathbf{x}$ for vectors and bold upper case letters $\mathbf{A}$ for matrices. All operations and functions are extended to sets in the usual way, e.g., for any number $x \in \mathbb{R}$ and sets $A, B \subseteq \mathbb{R}$, we have $x \cdot A = \{x \cdot a \mid a \in A\}$ and $A + B = \{a + b \mid a \in A, b \in B\}$. We use standard asymptotic notation $O(\cdot), o(\cdot), \omega(\cdot), \Omega(\cdot)$ to describe the order of growth of functions. Namely, $f = O(g)$ or $g = \Omega(f)$ means that $f(n) \leq c \cdot g(n)$ for some c and all sufficiently large n, and $f = \omega(g)$ or $g = o(f)$ means that $\lim_{n\to\infty} g(n)/f(n) = 0$.

2 Point Lattices: a primer

Lattices are regular arrangements of points in n-dimensional Euclidean space. The simplest example of a lattice is $\mathbb{Z}^n$, the set of all n-dimensional vectors with integer entries. In general, a lattice $L \subset \mathbb{R}^n$ is a set of points

$$L = \mathbf{B}\mathbb{Z}^k = \{\mathbf{Bx} \colon \mathbf{x} \in \mathbb{Z}^k\} \subset \mathbb{R}^n$$

obtained by applying a non-singular linear transformation $\mathbf{B} \in \mathbb{R}^{n\times k}$ to the integer lattice $\mathbb{Z}^k$. (See Figure 1 for a 2-dimensional example.)

The matrix $\mathbf{B} \in \mathbb{R}^{n\times k}$ is called a *lattice basis*, and *non-singular* means that its k columns should be linearly independent. For simplicity, in these notes, we focus on *full dimensional* lattices, i.e., lattices where $n = k$, and the lattice $\mathbf{B}\mathbb{Z}^n$ is generated by a square matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$ with nonzero determinant.

The lattice $\mathbf{B}\mathbb{Z}^n$ generated by a basis $\mathbf{B}$ can be equivalently written as the set $L = \{\sum_{i=1}^n \mathbf{b}_i \cdot x_i \colon \forall i. x_i \in \mathbb{Z}\}$ of all integer linear combinations of the basis vectors $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$. Notice the difference between a lattice and a vector space spanned by a basis $\mathbf{B}$: the vector space spanned by $\mathbf{B}$ is the set of all *real* linear combinations $\sum_i \mathbf{b}_i \cdot x_i$ (with $x_i \in \mathbb{R}$) of the basis vectors, while in the case of lattices we only take combinations with integer coefficients. A direct consequence of the definition (very easy to visualize, and only slightly harder to prove) is that lattices are discrete sets: the points in a lattice cannot be arbitrarily close to each other, i.e., for every lattice L there is an $\epsilon > 0$ such that the distance between any two distinct lattice points is at least ϵ. Due to this discrete structure, problems on lattices cannot generally be solved by simple linear algebra. For example, a basic fact from linear algebra is that any vector space has an orthogonal basis, and orthogonal bases are useful to solve a host of other problems. This is not true for lattices: not every lattice admits an orthogonal basis (e.g., see the lattice in Figure 1,) and this is what makes many lattice problems computationally hard to solve, and useful for cryptography.

[1] To be clear, all pictures you will ever see (e.g., in the slides that accompany these lecture notes, available from the author's web page at `http://www.cse.ucsd.edu/users/daniele`) describe 2-dimensional, or in some rare cases 3-dimensional lattices, while for lattice cryptography to be secure one needs to work with lattices in high dimensional space. Still, 2-dimensional lattices often provide good examples to illustrate the geometry of the problems.

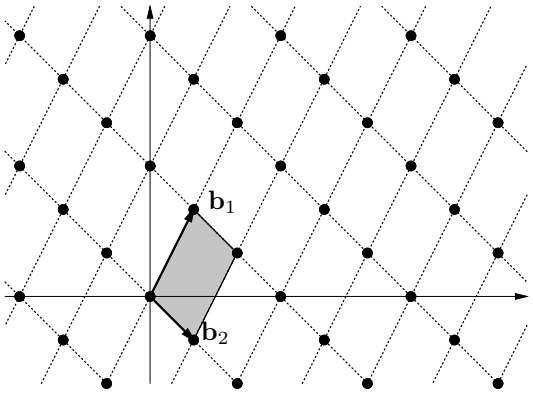

Figure 1: A 2-dimensional lattice $\mathbf{B} \cdot \mathbb{Z}^2$ generated by the basis $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2]$. The shaded areas is the parallelepiped $\mathcal{P}(\mathbf{B}) = \mathbf{B} \cdot [0,1)^2$ spanned by the basis vectors. The determinant of the lattice is the volume (area) of the parallelepiped.

The discrete structure of lattices yields also a more abstract (basis independent) characterization of lattices: a lattice is a *discrete additive subgroups* of $\mathbb{R}^n$.[2] We recall that a subgroup is a set which is closed under addition and subtraction.[3] Not every subgroup of $\mathbb{R}^n$ is a lattice, i.e., $\mathbb{R}^n$ itself is not a lattice because it is not discrete. In computer science and cryptography, it is common to focus on integer lattices, i.e., lattices $L \subseteq \mathbb{Z}^n$ whose vectors have all integer coordinates. Any set of integer vectors is certainly discrete, so integer lattices can be characterized simply as additive subgroups $\mathbb{Z}^n$.

2.1 Fundamental parameters

As n-dimensional objects, lattices can be hard to visualize, but there are several geometric quantities naturally associated to any lattice that succinctly describe some of its most important properties. These include

- The *determinant* of the lattice, which is the volume of the n-dimensional parallelepiped $\mathcal{P}(\mathbf{B}) = \mathbf{B}[0,1)^n$ spanned by the basis vectors $\det(\mathbf{B}\mathbb{Z}^n) = vol(\mathcal{P}(\mathbf{B}))$. (See Figure 1.) Intuitively, the determinant of a lattice describes the (inverse) density of lattice points in space, i.e., the number of lattice points in a sufficiently *large* and *regular* region of space $S \subseteq \mathbb{R}^n$, is approximately equal to $|S \cap L| \approx vol(S)/\det(L)$. To see this, notice that the space $\mathbb{R}^n$ can be tiled with copies of the fundamental parallelepiped $\mathcal{P}(\mathbf{B})$, with one copy $\mathbf{x} + \mathcal{P}(\mathbf{B})$ for every lattice point $\mathbf{x} \in L = \mathbf{B}\mathbb{Z}^n$. So, there is precisely one lattice point in every region of space $\mathbf{x} + \mathcal{P}(\mathbf{B})$ of volume $\det(L) = vol(\mathcal{P}(\mathbf{B}))$. For full dimensional lattices, the lattice determinant simply equals the absolute value of the matrix determinant $\det(\mathbf{B}\mathbb{Z}^n) = |\det(\mathbf{B})|$. In general, when $\mathbf{B} \in \mathbb{R}^{n \times k}$ for $k \leq n$, the determinant can be computed using the formula $\det(\mathbf{B}\mathbb{Z}^k) = \sqrt{\mathbf{B}^T\mathbf{B}}$. It is sometime convenient to consider the *root determinant* of a lattice, which is just the nth root of the determinant $\det(\mathbf{B}\mathbb{Z}^n)^{1/n}$.
- The *minimum distance* of a lattice, i.e., the smallest distance between any two lattice points $\lambda(L) = \inf\{\|\mathbf{x} - \mathbf{y}\| \mid \mathbf{x}, \mathbf{y} \in L, \mathbf{x} \neq \mathbf{y}\}$. The minimum distance is always attained by some pair of lattice points, so the *infimum* in the previous formula is in fact a *minimum*.
- More generally, one can define the *successive minima* of a lattice $\lambda_1, \ldots, \lambda_n$, where for every $i = 1, \ldots, n$, $\lambda_i(L)$ is the smallest positive real r such that the ball $\mathcal{B}(\mathbf{0}, r) = \{\mathbf{x} \colon \|\mathbf{x}\| \leq r\}$ of radius r centered at

[2]The fact that any such discrete subgroup admits a basis, and therefore is a lattice, is not at all obvious. But for the lattices considered in this survey, the proof is relatively easy and it is left to the reader as an exercise.

[3]Notice that requiring closure under subtraction is enough because $x + y = x - ((y - y) - y)$, and closure under addition is implied.

the origin contains at least i *linearly independent* vectors. As a special case, the first minimum $\lambda_1(L)$ is the length of the shortest nonzero lattice vector and equals the minimum distance λ of the lattice.

- The *covering radius* of a lattice $\rho(L)$ is the smallest real r number such that spheres of radius r centered around all lattice points cover the entire space (spanned by the lattice).

As useful way to visualize the minimum distance of a lattice is in terms of sphere packings. Think of putting equal spheres (of radius r) around every point in a lattice. When the radius r is very small, the spheres are disjoint. Now make the radius r as large as possible, but still subject to the constraint that the spheres centered around lattice points do not intersect. The maximum value of r is called the *packing radius* of the lattice, and it clearly equals half the minimum distance $\lambda(L)/2$. If we keep increasing the radius of the spheres beyond the packing radius $\lambda/2$, the spheres will start intersecting with each other, but they may not cover the entire space, leaving holes to be filled. But at some point, when r reaches the covering radius of the lattice ρ, all holes will be filled and the spheres will cover the entire space.

All these quantities are lattice invariant, i.e., they depend only on the set of lattice points $\mathbf{B}\mathbb{Z}^n$, and not on the specific basis $\mathbf{B}$ used to represent them. All these quantities are also invariant under rotations, i.e., for any orthogonal matrix $\mathbf{Q} \in \mathbb{R}^n$ we have $\lambda(L) = \lambda(\mathbf{Q}L)$, and similarly for the successive minima, covering radius, determinant and root determinant. All these quantities, with the exception of the determinant, are also linear in the sense that $\lambda(c \cdot L) = |c| \cdot \lambda(L)$ for any real scaling factor c, and similarly for λ_i, ρ and the root determinant.

Taken together, all these parameters provide a lot of useful information about the geometry of the lattice. The diligent reader can refine his geometric understanding of lattices by proving the following simple statements.

Exercise 1 *Show that for any lattice L and target point $\mathbf{t}$ (in the linear span of the lattice) there is* at most *one lattice point within distance (strictly less than) $\lambda(L)/2$ from $\mathbf{t}$.*

Exercise 2 *Show that for any lattice L and target point $\mathbf{t}$ (in the linear span of the lattice) there is* at least *one lattice point within distance $\rho(L)$ from $\mathbf{t}$.*

Exercise 3 *Show that for any n-dimensional lattice, the parameters $\lambda_1, \ldots, \lambda_n, \rho$ are related by the chain of inequalities*[4]

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n \leq 2\rho \leq \sqrt{n} \cdot \lambda_n.$$

Far less trivial to prove (but still not too hard, e.g., see [60] for a simple proof) is the following upper bound on the successive minima in terms of the (root-)determinant of the lattice.

Theorem 1 (Minkowski) *For any lattice Λ,*

$$\lambda(L) \leq \left(\prod_{i=1}^{n} \lambda_i(L)\right)^{1/n} \leq \sqrt{\gamma_n} \cdot \det(L)^{1/n}$$

where $\gamma_n \leq n$ is a function of the dimension only, and does not depend on the specific lattice.

For every n, the optimal value of γ_n in Minkowski's theorem is called *Hermite's constant* in dimension n. The exact value of γ_n is known only for $n = 1, 2, \ldots, 8$ and $n = 24$, but the asymptotic estimate $\gamma_n = \Theta(n)$ is all that is needed to understand lattice cryptography. Minkowski's bound provides useful information about the minimum distance of a lattice and it is pretty close to optimal when the lattice is chosen at random according to many natural probability distributions on lattices. (At this point the reader may want to try to come up with an explicit construction of an n-dimensional lattice with $\lambda(L) = \Omega(\sqrt{n}\det(L)^{1/n})$. It is not an easy task, and as we will see these are good examples of "cryptographically hard" lattices. See Exercise 13.) On the other hand, the actual minimum distance λ (or the mean value $(\prod_i \lambda_i)^{1/n}$) can be arbitrarily smaller than Minkowski's upper bound.

[4]The last two inequalities are not quite trivial. As a starter, the reader may want first to prove the weaker inequalities $\lambda_n \leq 2n\rho \leq n^2\lambda_n$.

Exercise 4 *Build a lattice basis* $\mathbf{B}$ *such that* $\lambda(\mathcal{L}(\mathbf{B}))$ *is smaller than Minkowski's upper bound by at least a factor* $c > 1$. [Hint: it is enough to consider lattices in dimension $n = 2$.]

On the other hand, for the mean value $(\prod_i \lambda_i)^{1/n}$, Minkowski's bound is loose by at most a $\sqrt{n}$ factor.

Exercise 5 *Show that for any* n*-dimensional lattice* L, $(\prod_i \lambda_i)^{1/n} \geq \det(L)^{1/n}$. [Hint: find a set of linearly independent vectors of length $\lambda_1, \ldots, \lambda_n$, and bound the volume of the parallelepiped they span using Hadamard's inequality.]

Another very useful relation between the fundamental lattice parameters λ and ρ is given by the so-called *transference theorems*, but in order to present them we need first to define the dual lattice. This is probably one of the more technical and less intuitive concepts used in the study of lattice cryptography, and it takes some time and effort to start developing some geometric intuition about it. But the notion of dual lattice turns out to be so useful that the effort is certainly justified. Let's put geometry aside for the moment. The reader is probably familiar with the definition of the dual of a vector space: the dual of a vector space $V \subseteq \mathbb{R}^n$ is the set $V^\dagger$ of all linear functions ϕ from V to $\mathbb{R}$. These functions can be naturally represented as vectors $\mathbf{x} \in V$, where $\phi_{\mathbf{x}}(\mathbf{v}) = \mathbf{v} \cdot \mathbf{x}$, and using this representation $V^\dagger$ is isomorphic to the original vector space V. The definition of dual lattice is analogous, but this time it is natural to consider linear functions $\phi \colon L \subseteq \mathbb{R}^n \to \mathbb{Z}$ that take integer values when evaluated on lattice points. As before, ϕ can be represented by a vector $\mathbf{x}$ in the linear span of L such that $\phi(\mathbf{v}) = \mathbf{v} \cdot \mathbf{x}$. The representation of linear functions as vectors leads to the following definition: the dual lattice $L^\dagger$ is the set of all vectors $\mathbf{x}$ (in the linear span of L) such that $\mathbf{v} \cdot \mathbf{x} \in \mathbb{Z}$ for every lattice point $\mathbf{v} \in L$. As you may expect, the dual lattice $L^\dagger$ is a lattice, and the dual of the dual is the original lattice $(L^\dagger)^\dagger = L$.

An initial understanding of the geometry of the dual lattice is provided by the following simple observations.

Exercise 6 *If* $\mathbf{B} \in \mathbb{R}^{n \times n}$ *is a lattice basis, then a basis of the dual lattice* $(\mathbf{B}\mathbb{Z}^n)^\dagger$ *is given by the inverse transpose matrix* $\mathbf{B}^{-T}$, *i.e.,* $(\mathbf{B}\mathbb{Z}^n)^\dagger = \mathbf{B}^{-T}\mathbb{Z}^n$. *More generally, the dual basis of* $\mathbf{B} \in \mathbb{R}^{n \times k}$, *is given by* $\mathbf{B}^\dagger = \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}$.

It easily follows that for any lattice L and scaling factor $c > 0$, the dual lattice of $c \cdot L$ is $(c \cdot L)^\dagger = (1/c) \cdot L^\dagger$, i.e., expanding a lattice by a factor c shrinks the dual lattice by the same factor. Also, we have that $\det(L^\dagger) = 1/\det(L)$. So, in an informal sense, the dual lattice is the "inverse" of the original lattice. The transference theorems show that the fundamental parameters λ and ρ are also connected by an inverse relation through the dual lattice, although only an approximate one.

Theorem 2 *For any* n*-dimensional lattice* L, *we have* $1 \leq \lambda_i(L) \cdot \lambda_{n-i+1}(L^\dagger) \leq n$ *and* $1 \leq 2\rho(L) \cdot \lambda(L^\dagger) \leq n$.

So, in an approximate sense (up to a factor n), $\lambda_1(L^\dagger), \ldots, \lambda_n(L^\dagger)$ are the inverses of $\lambda_n(L), \ldots, \lambda_1(L)$, and $\lambda(L^\dagger)$ is the inverse of $2\rho(L)$. This reinforces the geometric intuition behind the informal statement that the dual lattice $L^\dagger$ is the inverse of L. To get a better geometric grasp of this statement, consider the following illustrative example. Let L is a lattice such that $L^\dagger$ contains a very short vector $\mathbf{x}$ of length $\|\mathbf{x}\| = \lambda^\dagger$. By definition, all lattice points in L have integer scalar product with $\mathbf{x}$. So, $\mathbf{x}$ can be used to partition L into layers $L_i = \{\mathbf{v} \in L \mid \mathbf{x} \cdot \mathbf{v} = i\}$. Clearly, the covering radius $\rho(L)$ must be at least as large as half the distance between these layers, but the distance between the layers is precisely $1/\|\mathbf{x}\| = 1/\lambda^\dagger$. Therefore $1/(2\lambda^\dagger) \geq \rho$, yielding the lower bound $2\lambda^\dagger \cdot \rho \geq 1$. The upper bound $2\rho \cdot \lambda^\dagger \leq n$ is far less trivial to prove, and requires the use of harmonic analysis techniques [11].

2.2 Computational Problems

We have seen how the determinant of a lattice $\mathbf{B}$ can be efficiently computed, e.g., by the formula $\det(L) = \sqrt{\det(\mathbf{B}^T \cdot \mathbf{B})}$. However, the same is not true for the minimum distance $\lambda(L)$, any of the successive minima λ_i or the covering radius ρ. In fact, all these parameters give raise to well known hard computational problems on lattices. The best polynomial time approximation algorithms to solve any of the problems described

below only achieve approximation factors $\gamma(n) = 2^{O(n \log\log n / \log n)}$ almost exponential in the dimension of the lattice. Lattice cryptography is based on the assumption that no efficient algorithm can achieve polynomial approximation factors $\gamma(n) = n^{O(1)}$, at least in the worst-case, i.e., no efficient algorithm can achieve $\gamma(n) = n^{O(1)}$ approximation factor for every possible input lattice. Worst-case hardness assumptions are supported by many NP-hardness results for lattice problems [4, 21, 15, 10, 55, 26, 27, 42, 43, 38], although these results only hold for much smaller (sub-polynomial) approximation factors. The main problem underlying lattice cryptography is called the *Shortest Independent Vectors Problem*, and it is the natural computational problem associated to the parameter λ_n. Cryptography is based on the hardness of solving this problem even approximately, as defined below.

Definition 1 *The Shortest Independent Vector Problem (*SIVP_c*) is the following: given an n-dimensional lattice L (usually represented by a basis $\mathbf{B}$), find n* linearly independent *lattice vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in L$ such that $\max_i \|\mathbf{b}_i\| \leq c \cdot \lambda_n(L)$.*

In the above definition, $c \geq 1$ is the approximation factor, and it is usually a function of the dimension n. The case $c = 1$ corresponds to solving the problem exactly, i.e., finding n linearly independent lattice vectors such that the quantity $\max_i \|\mathbf{b}_i\|$ is as small as possible. We remark that n linearly independent lattice vectors are not necessarily a basis for the given lattice. For example, the column vectors $2\mathbf{B}$ are linearly independent vectors in $\mathbf{B}\mathbb{Z}^n$, but they are not a basis for $\mathbf{B}\mathbb{Z}^n$. This is because they only generate the sublattice $2 \cdot L$ rather than the entire lattice $L = \mathbf{B}\mathbb{Z}^n$. It is usually surprising to the novice to learn that there are lattices L such that any basis must necessarily contain vectors much longer than λ_n. In particular, there are lattices such that any solution to the SIVP problem will not be a basis for the lattice.

Exercise 7 *Let $L \subset \mathbb{Z}^n$ be the set of all integer vectors such that in each vector all coordinates have the same parity. (I.e., the coordinates are either all even, or all odd integers.) Assume $n \geq 5$. Prove that $\lambda_1 = \lambda_2 = \cdots = \lambda_n = 2$. Show that $L = \mathbf{B}\mathbb{Z}^n$ has no basis such that $\max_i \|\mathbf{b}_i\| \leq \lambda_n$.*

If you think this is surprising and counter intuitive, it is not for no reason. In dimension (up to) $n = 3$, a set of linearly independent vectors of length $\|\mathbf{b}_i\| = \lambda_i$ (for $i = 1, 2, 3$) is always a lattice basis. Only in dimension 4, they may or may not generate the entire lattice, and as the previous exercise shows, starting in dimension 5, minimizing the length of the vectors may necessarily yield only a basis for a sublattice. So, in order to "see" all this, you need to "look" beyond our 3-dimensional physical world. Fortunately, in most applications (and in particular, in cryptography), a set of short linearly independent vectors is just as good as a basis, and the reader can disregard the subtle distinction between a short basis, and a set of short linearly independent vectors. Moreover, if a basis is really needed, one can always convert a set of linearly independent vectors into a basis of slightly longer vectors.

Exercise 8 *Show that for any set of n linearly independent lattice vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n \in L$ there is a basis for $L = \mathbf{B}\mathbb{Z}^n$ such that $\max_i \|\mathbf{b}_i\| \leq \sqrt{n} \max_i \|\mathbf{v}_i\|$. Moreover, such $\mathbf{B}$ can be efficiently computed from $\mathbf{v}_1, \ldots, \mathbf{v}_n$ and an arbitrary basis for L.*

Another fundamental computational problem on point lattices is the Closest Vector Problem, which may be formulated as follows.

Definition 2 *Given a lattice $L \subset \mathbb{R}^n$, a target point $\mathbf{t} \in \mathbb{R}^n$, and a distance bound d, the Closest Vector Problem (*CVP*) asks for a lattice point $\mathbf{v} \in L$ at distance $\|\mathbf{t} - \mathbf{v}\| \leq d$ from the target, provided such a lattice point exists. In the exact version of* CVP*, the distance bound is taken to be the distance $d = \mu(\mathbf{t}, L) = \min_{\mathbf{v} \in L} \|\mathbf{t} - \mathbf{v}\|$ between the target and the lattice. In the approximate problem CVP_γ one sets $d = \gamma \cdot \mu$. Two special versions of* CVP *that play a prominent role in cryptography are:*

1. *the Bounded Distance Decoding problem (*BDD*), where $d < \lambda/2$, and*
2. *the Absolute Distance Decoding problem (*ADD*), where $d \geq \rho$.*

The importance of these specific parameter settings is that when $d < \lambda/2$, if a solution exists, then it is unique. (See Exercise 1.) On the other hand, when $d \geq \rho$, a solution is always guaranteed to exist (for any target $\mathbf{t}$), and it is generally not unique. (See Exercise 2.) Easier variants of BDD and ADD are obtained by introducing a slackness factor $\gamma \geq 1$, and strengthening the constrains on d to $d < \lambda/(2\gamma)$ (for BDD_γ) and $d \geq \gamma\rho$ (for ADD_γ). Informally, BDD_γ is the problem of finding the lattice vector closest to a target when the target is very close to the lattice, while ADD_γ is the problem of finding a lattice vector not too far from the target, where far is measured with respect to the absolute bound ρ within which a solution is always guaranteed to exist. As usual, the approximation factor γ can be a function of the dimension.

The problems SIVP, ADD and BDD are connected by some very interesting relations, which also play an important role in cryptography. To start with, observe that if the lattice is $L = \mathbb{Z}^n$, then CVP can be solved exactly and efficiently by rounding each coordinate of the target $\mathbf{t}$ to the closest integer, and this always result in a lattice vector within distance $\sqrt{n}/2$ from the target. More generally, for an arbitrary lattice L, we can attempt to solve ADD or BDD as follows. Let $\mathbf{B}$ be a lattice basis and think of $L = \mathbf{B}\mathbb{Z}^n$ as the integer lattice $\mathbb{Z}^n$ distorted by the linear transformation defined by $\mathbf{B}$. In order to find a lattice point close to a target $\mathbf{t}$ we may

1. first apply the inverse transformation $\mathbf{B}^{-1}$ to get $\mathbf{B}^{-1}\mathbf{t}$,
2. round $\mathbf{B}^{-1}\mathbf{t}$ to the closest integer vector $\lfloor \mathbf{B}^{-1}\mathbf{t} \rceil \in \mathbb{Z}^n$,
3. map the the resulting integer vector to the lattice point $\mathbf{v} = \mathbf{B}\lfloor \mathbf{B}^{-1}\mathbf{t} \rceil$.

The quality of this rounding procedure can be analyzed in terms of the two quantities

$$\begin{aligned} s_{\min}(\mathbf{B}) &= \min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{B}\mathbf{x}\|/\|\mathbf{x}\| \\ s_{\max}(\mathbf{B}) &= \max_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{B}\mathbf{x}\|/\|\mathbf{x}\| \end{aligned}$$

which express by how much the transformation $\mathbf{B}$ can shrink or expand the length of a vector.

Exercise 9 *Show that the above rounding procedure always returns a lattice point within distance* $\sqrt{n} \cdot s_{\max}(\mathbf{B})/2$ *from* $\mathbf{t}$. *Moreover, if* $\mathbf{t}$ *is within distance* $s_{\min}(\mathbf{B})/2$ *from the lattice, then the rounding procedure finds the (necessarily unique) lattice point within distance* $s_{\min}(\mathbf{B})/2$ *from* $\mathbf{t}$.

So, in order to solve ADD (resp. BDD) we need to find a basis $\mathbf{B}$ such that $s_{\max}(\mathbf{B})$ is small (resp. $s_{\min}(\mathbf{B})$ is large). This can be done by solving SIVP_γ, either in the lattice L or in its dual $L^\dagger$. Let's look first at ADD. Using an SIVP_γ oracle we can find a basis $\mathbf{B}$ with $\max_i \|\mathbf{b}_i\| \leq \sqrt{n}\gamma\lambda_n \leq 2\sqrt{n}\gamma\rho$. (See Exercises 3 and 8.) It is also easy to check that $s_{\max}(\mathbf{B}) \leq \sqrt{n}\max_i \|\mathbf{b}_i\| \leq 2\gamma n\rho$. This yields a solution to $\mathrm{ADD}_{n^{1.5}\gamma}$. In fact, by using a set of linearly independent vectors rather than a basis, in conjunction with a better rounding procedure based on orthogonalized projections, this can be improved to $\mathrm{ADD}_{\gamma\sqrt{n}}$.

Theorem 3 *For any* $\gamma \geq 1$, *there is a polynomial time (dimension preserving) reduction from* $\mathrm{ADD}_{\gamma\sqrt{n}}$ *to* SIVP_γ.

Let's now turn to BDD. This time we want $s_{\min}(\mathbf{B})$ to be large. Since $s_{\min}(\mathbf{B}) = 1/s_{\max}(\mathbf{B}^\dagger)$, this can be achieved by computing a short dual basis $\mathbf{B}^\dagger$ using SIVP_c. As before, the basis satisfies $s_{\max}(\mathbf{B}^\dagger) \leq 2\gamma n\rho(L^\dagger)$, and therefore by the transference theorem $s_{\min}(\mathbf{B}) \geq 1/(2\gamma n\rho^\dagger) \geq \lambda/(\gamma n^2)$. This yields a solution to $\mathrm{BDD}_{\gamma n^2}$. As before, this can be improved to $\mathrm{BDD}_{\gamma n}$ using a better rounding method, and further improvements are probably possible.

Theorem 4 *For any* $\gamma \geq 1$, *there is a polynomial time (dimension preserving) reduction from* $\mathrm{BDD}_{\gamma n}$ *to* SIVP_γ.

In summary, Theorems 3 and 4 show that SIVP is the hardest of the three problems, in the sense that both ADD and BDD can be efficiently solved given an SIVP oracle, at least approximately, up to polynomial approximation factors.

In fact, it is not hard to show that $\mathrm{SIVP}_{\gamma\sqrt{n}}$ reduces to ADD_γ. (This is implicit in the proof of the last two inequalities in Exercise 3.) So, ADD and SIVP are equivalent problems, up to small polynomial factors in the quality of the solution. Interestingly, BDD seems an easier problem than ADD and SIVP: BDD reduces to ADD and SIVP, but in the opposite direction reductions are known only under quantum algorithms! Giving a (classical, possibly randomized, but not quantum) reduction from ADD or SIVP to BDD is an important open problem in the complexity of lattice problems, and a problem that, as we will see later, has special cryptographic significance.

We conclude this section with a brief mention of the standard CVP_γ and the Shortest Vector Problem SVP_γ, which is the problem of finding a nonzero lattice vector of length at most $c\lambda_1$. These are the two most famous computational problems in the algorithmic study of point lattices. They are also the hardest: they are equivalent to each other (up to polynomial approximation factors), and SIVP reduces to them. However, no reduction (even quantum) is known in the opposite direction. In particular, it is not known how to build cryptographic function based on the conjectured (worst-case) intractability of SVP or CVP. Therefore, another very important open problem in the complexity of lattice problems is to give a reduction from SVP or CVP to SIVP.

3 Random Lattices and Lattice Cryptography

Lattice cryptography is based on the conjectured hardness of SIVP, i.e., the problem of computing (a set of n linearly independent) short vectors in a lattice. Intuitively, one can think of building a cryptosystem where public keys are lattices (e.g., represented by a basis consisting of long vectors), with the corresponding secret key given by the short lattice vectors. If finding short vectors is hard, then the secret key cannot be easily computed from the public description of the lattice. Moreover, the public/secret key pair can be used to encrypt/decrypt messages as follows.[5]

- Let $L = \mathbf{B}\mathbb{Z}^n$ be a lattice and $\mathbf{S}$ a set of linearly independent lattice vectors, serving respectively as the public and secret key of the cryptosystem.
- A message m is encrypted by encoding it as a short random vector $\mathbf{x}$, selecting a "random" point $\mathbf{v} \in \mathbf{B}^\dagger\mathbb{Z}^n$ in the dual lattice, and perturbing $\mathbf{v}$ by $\mathbf{x}$, to yield ciphertext $\mathbf{c} = \mathbf{v} + \mathbf{x}$. Notice that the ciphertext is a point within distance $\|\mathbf{x}\|$ from the dual lattice $L^\dagger$. The reason we use a dual lattice vector here is that a good basis $\mathbf{S}$ of L allows to solve BDD in $L^\dagger$.
- If $\|\mathbf{x}\| < \lambda(L^\dagger)/2$, then recovering $\mathbf{x}$ gives raise to a BDD problem. As seen in Theorem 4, the secret key $\mathbf{S}$ can be used to find the lattice point $\mathbf{v}$ closest to the target ciphertext $\mathbf{c}$, provided the error vector $\mathbf{x}$ is short enough. The message is then recovered as $\mathbf{x} = \mathbf{c} - \mathbf{v}$.

In summary, recovering the secret key $\mathbf{S}$ from the public description of the lattice $\mathbf{B}$ corresponds to a SIVP instance, while decrypting without the knowledge of the secret key is an instance of BDD. Of course, this is very informal, and leaves many important questions unanswered. What basis $\mathbf{B}$ should be used as the public key? How should the "random" lattice point $\mathbf{v} \in L^\dagger$ be selected? What probability distribution should be used/assumed on the error vector $\mathbf{x}$? In what sense is the above scheme secure? For many of these questions, provably optimal answers are known [54]: both integer lattices $L^\dagger \subseteq \mathbb{Z}^n$ and their cosets $\mathbf{x} + L^\dagger$ admit easily computable normal forms, or standard representatives.[6] Since these standard representatives can be easily

[5] Here we treat encryption somehow informally, like textbook RSA, where the message is assumed to be somehow random, and we identify cryptosystems with the theoretical notion of trapdoor function families. The reader familiar with modern cryptography will certainly know that for cryptosystems to provide a reasonable level of security, they need to be randomized. The example here is only meant to provide the intuitive geometric intuition of how lattice cryptography works, without getting into technical details.

[6] For example, given a lattice basis $\mathbf{B}$, a standard representative of the coset $\mathbf{x}+\mathbf{B}\mathbb{Z}^n$ is given by the unique point of $\mathbf{x}+\mathbf{B}\mathbb{Z}^n$ inside the parallelepiped $\mathcal{P}(\mathbf{B})$. For the basis $\mathbf{B}$ one may use the Hermite Normal Form [54].

computed from any other representation of the lattice and its cosets, they provide the minimum possible amount of information to the adversary, and therefore the highest level of security. But what distribution $\mathbf{x}$ should be used to encode the messages, and, more importantly, what distribution should be used for the lattices $L, L^\dagger$ to ensure that SIVP (in L) and BDD (in $L^\dagger$) are computationally hard?

The conjectured (worst-case) intractability of SIVP, ADD and BDD is not directly relevant here. In cryptography one needs computational problems that are hard to solve on the average: when you pick the secret key of a cryptographic function at random (according to an appropriate probability distribution), it is not enough to know that *some* key is computationally hard to break. You want to be confident that *your* randomly chosen key is hard to break with high probability. So, in order to formalize lattice cryptography, we need to introduce a notion of *random* lattice, so that when L is chosen according to this random distribution, SIVP on L is hard.

3.1 Random lattices

The two most common distributions over n-dimensional lattices encountered in cryptography are defined as follows. Fix positive integers $k \leq n \leq q$, where k serves as the main security parameter. Typically n is a small multiple of k (e.g., $n = O(k)$ or $n = O(k \log k)$) and q is a small prime with $O(\log k)$ bits. Notice that q is very small, not at all like the large primes (with $O(k)$ bits) used in number theoretic cryptography. For any matrix $\mathbf{A} \in \mathbb{Z}_q^{k \times n}$ define

$$\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^n : \mathbf{A}\mathbf{x} = \mathbf{0} \bmod q\}$$

$$\Lambda_q(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^n : \mathbf{x} = \mathbf{A}^T\mathbf{s} \bmod q \text{ for some } \mathbf{s} \in \mathbb{Z}_q^k\}.$$

Intuitively, $\Lambda_q(\mathbf{A})$ is the lattice generated by the rows of $\mathbf{A}$ modulo q, while $\Lambda_q^\perp(\mathbf{A})$ is the set of solutions of the system of k linear equations modulo q defined by the rows of $\mathbf{A}$. It is easy to check that $\Lambda_q(\mathbf{A})$ and $\Lambda_q^\perp(\mathbf{A})$ are subgroups of $\mathbb{Z}^n$, and therefore they are lattices. It is also apparent from the definition that these lattices are q-ary, i.e., they are periodic modulo q: one can take the finite set Q of lattice points with coordinates in $\{0, \ldots, q-1\}$, and recover the whole lattice by tiling the space with copies $Q + q\mathbb{Z}^n$. The matrix $\mathbf{A}$ used to represent them is not a lattice basis. A lattice basis $\mathbf{B}$ for the corresponding lattices can be efficiently computed from $\mathbf{A}$ using linear algebra, but it is typically not needed: cryptographic operations are usually expressed and implemented directly in terms of $\mathbf{A}$. A random lattice is obtained by picking $\mathbf{A} \in \mathbb{Z}_q^{k \times n}$ uniformly at random. The corresponding distributions are denoted $\Lambda_q(n,k)$ and $\Lambda_q^\perp(n,k)$.

Regarding the error vector distribution, one possible way to choose $\mathbf{x}$ may be to select it uniformly at random among all integer vectors of bounded norm, but for technical reasons a different distribution is often more convenient. In lattice cryptography, perturbation vectors are typically chosen according to the Gaussian distribution D_α which picks each $\mathbf{x} \in \mathbb{Z}^n$ with probability (roughly) proportional to $\exp(-\pi\|\mathbf{x}/\alpha\|^2)$. The Gaussian distribution has the analytical advantage that the probability of a point $\mathbf{x}$ depends only on its norm $\|\mathbf{x}\|$, and still the coordinates of $\mathbf{x}$ can be chosen *independently* (namely, each with probability proportional to $\exp(-\pi|x_i/\alpha|^2)$). It can be shown that when $\mathbf{x}$ is chosen according to this distribution (over $\mathbb{Z}^n$), $\Pr\{\|\mathbf{x}\| > \sqrt{n}\alpha\}$ is exponentially small. So, by truncating a negligibly small tail, D_α can be regarded as a probability distribution over the integer vectors of norm bounded by $\alpha\sqrt{n}$.

Before providing a theoretical justification for using these distributions in cryptography, let us try to get a better understanding of the lattices. We begin by observing that $\Lambda_q(\mathbf{A})$ and $\Lambda_q^\perp(\mathbf{A})$ are dual to each other, up to a scaling factor q.

Exercise 10 *Show that* $\Lambda_q(\mathbf{A}) = q \cdot \Lambda_q^\perp(\mathbf{A})^\dagger$ *and* $\Lambda_q^\perp(\mathbf{A}) = q \cdot \Lambda_q(\mathbf{A})^\dagger$. *In particular,* $\det(\Lambda_q(\mathbf{A})) \cdot \det(\Lambda_q^\perp(\mathbf{A})) = q^n$. *Moreover, for any* $\mathbf{A} \in \mathbb{Z}_q^{k \times n}$, *we have* $\det(\Lambda_q^\perp(\mathbf{A})) \leq q^k$ *and* $\det(\Lambda_q(\mathbf{A})) \geq q^{n-k}$.

To better understand the relation between $\Lambda_q(n,k)$ and $\Lambda_q^\perp(n,k)$, it is convenient to define two auxiliary distributions. Let $\tilde{\Lambda}_q^\perp(n,k)$ the conditional distribution of a lattice chosen according to distributions $\Lambda_q^\perp(n,k)$, given that the lattice has determinant exactly q^k. Similarly, let $\tilde{\Lambda}_q(n,k)$ be the conditional distribution of a lattice chosen according to $\Lambda_q(n, n-k)$, given that the determinant of the lattice is q^{n-k}. In both cases, when

q is a prime, the condition is equivalent to requiring that the rows of $\mathbf{A}$ are linearly independent modulo q. How much do these conditional distributions differ from the original ones? Not much.

Exercise 11 *Show that if* $\mathbf{A} \in \mathbb{Z}_q^{k\times n}$ *is chosen uniformly at random, then* $\Pr\{\det(\Lambda_q^\perp(\mathbf{A})) = q^k\} = \Pr\{\det(\Lambda_q(\mathbf{A})) = q^{n-k}\} \geq 1 - 1/q^{n-k}$. *Moreover, the conditional distributions* $\tilde{\Lambda}_q^\perp(n,k) = \tilde{\Lambda}_q(n,n-k)$ *are identical.*

So, for typical settings of the parameters (e.g., $n \geq 2k$), lattices chosen according to $\Lambda_q^\perp(n,k)$ or $\Lambda_q(n,n-k)$ have determinant q^k except with negligible probability $\epsilon \leq q^{-k}$, and the distributions $\Lambda_q^\perp(n,k)$ and $\Lambda_q(n,n-k)$ are almost identical because they are both statistically close to $\tilde{\Lambda}_q^\perp(n,k) = \tilde{\Lambda}_q(n,n-k)$.

We now move to estimating the parameters of lattices chosen according to these distributions. Clearly, we always have $\lambda_1 \leq \lambda_n \leq q$ and $\rho \leq \sqrt{n}q$ because $q\mathbf{I}$ gives a set of n linearly independent vectors of length q. In fact, from Theorem 1 and Exercise 10, we know that $\lambda(L) \leq \sqrt{n}q^{k/n}$ for any $L \in \Lambda_q^\perp(n,k)$. This upper bound is essentially tight.

Exercise 12 *There is a constant* $\delta > 0$ *such that if* L *is chosen according to* $\Lambda_q^\perp(n,k)$, *then* $\Pr\{\lambda(L) < \delta\sqrt{n}q^{k/n}\} \leq 1/2^n$. [Hint: consider all integer vectors of norm at most $\delta\sqrt{n}q^{k/n}$ and use a union bound.]

What about the other parameters λ_n, ρ? Also these parameters are very close to Minkowski's upper bound with high probability.

Exercise 13 *If* L *is chosen according to* $\Lambda_q^\perp(n,k)$, *then* $\Pr\{\lambda(L) \leq 2\rho(L) < \frac{1}{\delta} \cdot \sqrt{n} \cdot q^{k/n}\} \leq 1/2^n$.[Hint: Prove the bound for $\Lambda_q(n,n-k) \approx \Lambda_q^\perp(n,k)$ instead, and use duality and the transference theorems.]

In summary, when a lattice is chosen according to $\Lambda_q^\perp(n,k) \approx \tilde{\Lambda}_q^\perp(n,k) = \tilde{\Lambda}_q(n,n-k) \approx \Lambda_q(n,n-k)$, all the parameters $\lambda_1, \ldots, \lambda_n, \rho$ are within a constant factor from Minkowski's bound $\sqrt{n}q^{k/n}$ with overwhelming probability. This provides very useful information about what it means to solve ADD or BDD on these random lattices. Let L be a lattice chosen according to $\Lambda_q^\perp(n,k)$, and let $\mathbf{t} = \mathbf{v} + \mathbf{x}$ be a lattice point $\mathbf{v} \in L$ perturbed by a noise vector $\mathbf{x}$ chosen according to distribution $D_{cq^{k/n}}$ over $\mathcal{B}(c\sqrt{n}q^{k/n})$. Finding a lattice point within distance $c\sqrt{n}q^{k/n}$ from $\mathbf{t}$ is an ADD problem when $c > \delta$ and it is a BDD problem when $c < 1/\delta$.

3.2 One way functions

We now show that solving ADD and BDD on random lattices can be formulated as the problem of inverting the function

$$f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \pmod{q}. \tag{1}$$

when the matrix $\mathbf{A} \in \mathbb{Z}_q^{k\times n}$ is chosen uniformly at random and the input is chosen according to distribution $D_{cq^{k/n}}$, for some $c > 0$. Of course, given a matrix $\mathbf{A}$ and a value $\mathbf{b} = f_{\mathbf{A}}(\mathbf{x})$, recovering a possible preimage of $\mathbf{y}$ under $f_{\mathbf{A}}$ is just a matter of performing some linear algebra, and it can be efficiently accomplished in a variety of ways. In order to get a hard-to-invert function from (1), one needs to regard $D_{cq^{k/n}}$ as a probability distribution over $\mathcal{B}(c\sqrt{n}q^{k/n})$, and consider $f_{\mathbf{A}}$ as a function with this domain. The relation between inverting $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{b}$ (i.e., finding small solutions to the inhomogeneous system $\mathbf{A}\mathbf{x} = \mathbf{b} \pmod{q}$) and lattice problems is easily explained. Using linear algebra, one can efficiently find an arbitrary solution $\mathbf{t} \in \mathbb{Z}_q^n$ to the system, but this solution will generally have large entries and not belong to the domain of $f_{\mathbf{A}}$. Linear algebra gives us a little more than an arbitrary solution. It tells us that any solution to the inhomogeneous system $\mathbf{A}\mathbf{x} = \mathbf{b}$ can be expressed as the sum of any fixed specific solution $\mathbf{t}$ to $\mathbf{A}\mathbf{t} = \mathbf{b} \pmod{q}$ and a solution $\mathbf{z}$ to the homogeneous system $\mathbf{A}\mathbf{z} = \mathbf{0} \pmod{q}$. But the set of solutions to the homogeneous system is precisely the lattice $\Lambda_q^\perp(\mathbf{A})$. So, finding a small $\mathbf{x} = \mathbf{t} + \mathbf{z}$ is equivalent to finding a lattice point $-\mathbf{z} \in \Lambda_q^\perp(\mathbf{A})$ within distance $\|\mathbf{t} - (-\mathbf{z})\| = \|\mathbf{x}\|$ from the target $\mathbf{t}$. In summary, inverting $f_{\mathbf{A}}$ is equivalent to the problem of finding lattice vectors in $\Lambda_q^\perp(\mathbf{A})$ within distance $c\sqrt{n}q^{k/n}$ from the target. Depending of the value of c, this is the ADD_c problem (for $c > \delta$) or the $\mathrm{BDD}_{1/c}$ problem (for $c < 1/\delta$).

These two different ranges for c also corresponds to very different statistical properties of the function $f_{\mathbf{A}}$. Namely, when $c < 1/\delta$, the function $f_{\mathbf{A}}$ is injective with high probability (which corresponds to BDD having at most one solution).

Exercise 14 *Show that for $c \leq \delta$ the function $f_{\mathbf{A}}$ with domain $\mathcal{B}(c\sqrt{n}q^{k/n})$ is injective with overwhelming probability over the choice of $\mathbf{A}$.*

Similarly, when $c > \delta$, the function $f_{\mathbf{A}}$ is surjective with high probability. In fact, all the output values in $\mathbb{Z}_q^n$ have almost the same probability under $f_{\mathbf{A}}(D_{cq^{k/n}})$.

Theorem 5 ([34]) *Let q be a prime, $n \geq 2k \log q$ and $c \geq \omega(\sqrt{\log n})$. Then, with overwhelming probability over the choice of $\mathbf{A} \in \mathbb{Z}_q^{k \times m}$, if $\mathbf{x}$ is chosen according to distribution $D_{c \cdot q^{k/n}}$, then $f(\mathbf{x})$ is statistically close to uniform over $\mathbb{Z}_q^k$.*

Now, let us go back to the question: are lattice problems hard on average when the lattice is chosen according to $\Lambda_q^{\perp}(n, k)$? A first answer to this question was given in a seminal paper of Ajtai [6] which marks the beginning of modern day lattice based cryptography. Ajtai's result has been subsequently improved and simplified in a sequence of papers [20, 56, 63]. The strongest known result is the one due to [63] with some additional refinements in [34].

Theorem 6 (Implicit in [63, 34]) *For any polynomially bounded $n(k)$ and $c(k) > \delta$, and any $\omega(k \log k) \leq q(k) \leq k^{O(n/k)}$, the following holds. Assume SIVP_γ is hard in the worst case (over k-dimensional lattices) for approximation factors $\gamma(k) = \omega(c\sqrt{n \cdot k \log k} \cdot q^{k/n})$. Then $f_{\mathbf{A}}$ is a one-way function with input distribution $D_{cq^{k/n}}$ (over $\mathcal{B}(c\sqrt{n}q^{k/n})$).*

Typically, $q(k) = k^{O(1)}$ is a small polynomial, $n = k \log q = O(k \log k)$, and $c > \delta$ a constant. With this parameters, Theorem 6 gives a provably secure one-way function under the assumption that SIVP_γ is hard to approximate (in the worst-case, over k-dimensional lattices) within factors $\gamma(k) = \omega(k \log k)$ almost linear in the dimension of the lattice.

Recall that the one-way function inversion problem from Theorem 6 is an average case version of ADD for lattices distributed according to $\Lambda_q^{\perp}(n, k)$. Also, in the previous section we have seen that SIVP is equivalent to ADD, up to small polynomial approximation factors. So, Theorem 6 can be interpreted as a connection between the worst-case and average-case complexity of ADD, and justifies using distribution $\Lambda_q^{\perp}(n, k)$ to select the lattice. If ADD is not hard on average for this distribution, then it is not hard at all!

What about BDD, i.e., inverting $f_{\mathbf{A}}(\mathbf{x})$ when the input $\mathbf{x}$ is distributed according to $D_{cq^{k/m}}$ for some small $c < 1/\delta$? For these inputs the behavior of the function $f_{\mathbf{A}}$ is quite different: we have seen that $f_{\mathbf{A}}$ is injective (with high probability) and therefore it is trivially collision resistant: there exist no $\mathbf{x} \neq \mathbf{y}$ such that $f_{\mathbf{A}}(\mathbf{x}) = f_{\mathbf{A}}(\mathbf{y})$. But for injective functions, collision resistance is a trivial property and does not imply any form of one-wayness. Still, it is possible to prove that $f_{\mathbf{A}}$ is a one-way function, using the stronger complexity assumption that SIVP is hard to solve in the worst case even by quantum algorithms. Since we are interested in solving BDD in the dual lattice, we state this result in terms of the (scaled) dual lattice distribution $q(\Lambda_q^{\perp}(n, k))^{\dagger} = \Lambda_q(n, k) \approx \Lambda_q^{\perp}(n, n - k)$.

Theorem 7 (Implicit in [73]) *For all $c < 1/\delta$, some polynomial $n(k) = k^{O(1)}$, and prime $q > (2\sqrt{k}/c)^{n/(n-k)}$, the following holds true. Assume SIVP_γ is hard in the worst case (over k-dimensional lattices) even for quantum algorithms, for approximation factor $\gamma(k) = \omega(\sqrt{k}) \cdot k \cdot q^{k/n}/c$. Then $f_{\mathbf{A}}$ is a one-way function with input distribution $D_{c \cdot q^{1-k/n}}$ and domain $\mathcal{B}(c\sqrt{n} \cdot q^{1-k/n})$.*

Notice that $D_{c \cdot q^{1-k/n}}$ is the natural distribution to use when the lattice is chosen according to $\Lambda_q^{\perp}(n, n-k)$ because these lattices have root determinant $q^{1-k/n}$ rather than $q^{k/n}$. Theorem 7 asserts that $f_{\mathbf{A}}$ is a one-way function only when $n(k)$ is some polynomially bounded, but unspecified, function of k. This is because the BDD problem in $\Lambda_q(n, k)$ lattices in [73] is described as a learning problem (the LWE problem, for Learning With Errors), where each coordinate of the lattice corresponds to a noisy sample, and the adversary is assumed to have access to an arbitrary (but polynomially bounded) number of samples. In principle, it

should be possible to tighten the proof in [73] to show that a relatively small value of $n(k)$ is sufficient for the reduction to go through. But, anyway, this is not the main difference if we compare the result with Theorem 6. A more fundamental difference between Theorem 6 and Theorem 7 is that the latter is based on a quantum reduction. The reason the power of quantum computers is needed for establishing the one-wayness of $f_{\mathbf{A}}$ in the BDD setting is currently one of the mysteries of lattice cryptography, and it reflects our poor understanding of the relation between ADD and BDD problems. Even in the classical setting of worst-case reductions, we know how to approximately reduce BDD to ADD, but no reduction is known in the opposite direction, at least classically. So, BDD seems an easier problem than ADD also in the worst-case. In the average case setting, Theorems 6 and 7 can be summarized as saying that ADD on random q-ary lattices is as hard as approximating SIVP with classical algorithms, while BDD on random q-ary lattices is as hard as approximating SIVP but with quantum algorithms, under which SIVP may be easier to solve.

We remark that for certain choice of parameters (e.g., when $q = 2^{O(n)}$ is exponentially large,) it is possible to show that the function $f_{\mathbf{A}}$ of Theorem 7 is one-way, assuming the classical worst-case hardness of some lattice problems [67]. The worst-case problems can be BDD, a special version of SVP with "unique solution", or the problem or approximating the value of λ without necessarily finding a short lattice vector. All this problems are essentially equivalent to each other [50] and conceivably easier than SIVP. So, in some settings, Theorem 7 can be interpreted as a connection between the worst-case and average-case complexity of BDD under classical algorithms.

In summary, assuming that SIVP is hard to solve in the worst case with classic algorithms, is enough to build simple cryptographic functions, like one-way functions, and as will will see in the next section, also collision resistant hash functions, and commitment schemes. The construction of public key encryption and more complex cryptographic primitives requires the stronger assumption that either BDD is hard in the worst-case (if an exponentially large value of q can be used) or SIVP and ADD are hard even under quantum algorithms. Providing a classical reduction from SIVP and ADD to BDD is an important open problem as it would unify the complexity assumptions used for the construction of all lattice cryptography primitives.

4 Applications

In this section we give some representative examples of cryptographic applications based on lattices. The presentation is informal, emphasizing the geometric ideas behind the constructions, rather than the technical details of the cryptographic definitions and security proofs. Here we aim primarily at illustrating the wide range of applications enabled by lattice cryptography, and highlight how cryptographic constructions, which are often described simply in terms of matrices and indistinguishability properties, often have a natural geometric interpretation. However, we remark that all the constructions discussed here can be properly analyzed and proved secure based on the conjectured one-wayness of the functions $f_{\mathbf{A}}$ discussed in Section 3, or, using Theorems 6 and 7, the worst-case hardness of SIVP under classical or quantum algorithms. For precise statements, security definitions and formal proofs, the reader is referred to the original papers, cited throughout this section.

4.1 Hash functions

Hash functions are (keyed) functions that compress long input strings, into shorter digests, and still have the property that they are collision resistant: it is computationally hard to find two distinct inputs $x \neq y$ such that $f(x) = f(y)$. For injective functions, this property holds trivially, so we may regard collision resistance as a computational relaxation of injectivity, and say that $f_{\mathbf{A}}$ is pseudo-injective.

Now, consider the function $f_{\mathbf{A}}$ for $\mathbf{A} \in \mathbb{Z}_q^{k \times m}$ and input $\mathbf{x} \leftarrow D_{cq^{k/n}}$. It turns out that (when $\mathbf{A}$ is chosen at random) the proof of Theorem 6 shows that the function $f_{\mathbf{A}}$ is not only one-way, but also collision resistant. By Exercise 14, when $c < 1$, the function $f_{\mathbf{A}}$ is injective with high probability, and collisions $f_{\mathbf{A}}(\mathbf{x}) = f_{\mathbf{A}}(\mathbf{y})$ do not exist. When c grows beyond 1, the function $f_{\mathbf{A}}$ is no longer injective over the domain associated to the input distribution. Still, as proved in [63, 34], the function $f_{\mathbf{A}}$ is collision resistant for the parameters

of Theorem 6. The intuitive explanation is that inverting $f_{\mathbf{A}}$ for these parameters corresponds to the ADD problem, while finding collisions corresponds to finding short vectors $\Lambda_q^{\perp}(\mathbf{A})$ of length $O(\sqrt{n}q^{k/m}) \approx \lambda_n$. Since ADD and SIVP are equivalent problems, we can expect that these two problems have roughly the same complexity. In this informal argument, we are hiding the fact that the equivalence between ADD and SIVP is proved for the worst-case formulation of the problems, and that SIVP requires to find not just one, but many short vectors. Still, the intuition holds, and $f_{\mathbf{A}}$ can be formally shown to be collision resistant, on the average, when $\mathbf{A}$ is chosen at random.

Hash functions are usually designed using ad-hoc methods, without supporting proofs of security. Lattice based functions like $f_{\mathbf{A}}$ are not competitive with such ad-hoc constructions. Still, using special classes of algebraic lattices it is possible to get much more efficient constructions [57, 48, 68, 51] which sill are provably hard to break under worst-case assumption (albeit over a restricted class of lattices) and at the same time are comparable to ad-hoc constructions in terms of efficiency.

4.2 Pseudorandom generators

When $c > 1$ is large enough, the output of $f_{\mathbf{A}}(\mathbf{x})$ with $\mathbf{x} \leftarrow D_{cq^{k/n}}$ is statistically close to random, and the function compresses its input. When $c < 1$ is small, the output of $f_{\mathbf{A}}(\mathbf{x})$ is certainly not close to uniform, just because the function stretches its input to a longer string. Still, the output of $f_{\mathbf{A}}(\mathbf{x})$ can be shown computationally indistinguishable from uniform, i.e., it is pseudo-random. In other words, $f_{\mathbf{A}}$ is a pseudo-random generator family, indexed by $\mathbf{A}$.

The pseudo-randomness of $f_{\mathbf{A}}(\mathbf{x})$ was proved in a loose sense already in [73]. The proof in [73] was loose in the sense that in order to prove the pseudo-randomness of the output of $f_{\mathbf{A}}$ for certain values of the parameters n, k, q, it required to assume that $f_{\mathbf{A}}$ was one-way for a (polynomially related, but) much bigger value of n. Recently [61] it has been shown that such increase is not necessary, and the pseudo-randomness of the output can be proved for the same values of the parameters as in Theorem 7.

We recall that inverting $f_{\mathbf{A}}$ for this range of parameters corresponds to a BDD problem. On the other hand, distinguishing the output of $f_{\mathbf{A}}$ from random corresponds to decisional version of BDD: the input to BDD is a point close to the lattice, while the uniform distribution in space can be shown to be far from the lattice with high probability [36, 37]. So, proving that the output of $f_{\mathbf{A}}$ is pseudo-random roughly corresponds to showing that if you can tell if a target is either very close, or very far from a lattice, then given a point very close to the lattice one can find the closest lattice point. For a reduction of this kind in the worst-case setting, see [45].

4.3 Commitment schemes

A (non-interactive) commitment scheme is a function that allows one party to commit to a value x while keeping it hidden. At a later time the same party can open the commitment, and reveal the value x. The commitment should be binding, i.e., it should be computationally unfeasible to commit to a value x and then open the commitment to a different value x'. The construction of commitment schemes from lattices is very simple: in order to commit to a value represented by a short vector $\mathbf{x}$, choose a short random vector $\mathbf{r}$, and output $f_{\mathbf{A}}(\mathbf{x}, \mathbf{r})$, where matrix $\mathbf{A}$ is wide enough that it can be multiplied by the concatenation of the two vectors $(\mathbf{x}; \mathbf{r})$. This commitment scheme was proposed in [41], where it is also used for the construction of a concurrently secure identification scheme based on lattices.

We remark that, depending on the value of the parameters, this construction gives both commitments that are statistically hiding and computationally binding, and also commitments that are computationally hiding, but statistically binding. The fact that this construction gives a secure commitment scheme is closely related to the fact that $f_{\mathbf{A}}$ is both (pseudo-)random and (pseudo-)injective. (Pseudo-)randomness of the output gives the hiding property of the commitment, while the (pseudo-)injective property implies that the commitment is binding. Depending on the setting of the parameter c in the input distribution $D_{cq^{k/n}}$, either the hiding or binding property (but not both) may hold in a strong statistical sense.

4.4 Public key encryption

A public key encryption scheme can be built along the lines described at the beginning of Section 3:

- The public key is a random lattice $\Lambda_q(\mathbf{A})$. The secret key is given by a short basis or set of short vectors $\mathbf{S}$.
- Ciphertexts are obtained by encoding a message into a short vector $\mathbf{x}$, and computing a standard representative for the coset $\mathbf{x} + (\Lambda_q(\mathbf{A}))^\dagger$. Such representative can be computed, for example, as $f_{\mathbf{A}}(\mathbf{x})$.
- Decryption is a BDD problem in the dual lattice $(\Lambda_q(\mathbf{A}))^\dagger$, and it can be performed using the short basis $\mathbf{S}$.

We already know that recovering the secret key from the public one corresponds to an SIVP problem in $\Lambda_q(\mathbf{A})$, which by Theorem 6 is hard on average. Similarly, decryption is an average-case BDD problem, and by Theorem 7 it is also hard. All we are missing is a method to generate a random lattice $\Lambda_q(\mathbf{A})$ together with a short set of lattice vectors $\mathbf{S}$.

In fact, we don't need a full set of n-linearly independent short lattice vectors in order to build a public key encryption scheme. Assume for simplicity we want to encrypt a single bit message. Then the message 0 may be encoded as a point close to the (dual) lattice, while 1 may be encoded as a random point, which will be far from the lattice with high probability. Then, we can decrypt using a single short (dual) vector $\mathbf{s}$ as follows: recall that a lattice vector $\mathbf{s} \in L$ can be used to partition the dual lattice into layers $L_i^\dagger = \{\mathbf{v} \in L^\dagger \mid \mathbf{v} \cdot \mathbf{s} = i\}$. One can check if a point $\mathbf{x}$ is either close or far from any of these layers by computing the product $\mathbf{x} \cdot s$ and checking if the result is close to an integer. This idea was already present in the cryptosystem of Ajtai and Dwork [7], which didn't use q-ary lattices.

So, how can we obtain this short vector $\mathbf{s}$ when the lattice $(\Lambda_q(\mathbf{A}))$ is chosen at random? Remember that finding short vectors in these random lattices is a hard problem, and it has to be as it corresponds to the key recovery problem of our cryptosystem. Fortunately we can *plant* $\mathbf{s}$ in the lattice during key generation process. We recall that $\Lambda_q(\mathbf{A})$ is the lattice generated by the columns of $\mathbf{A}^T$ modulo q. So, we can choose the first $k-1$ columns of $\mathbf{A}^T = [\mathbf{A}_0^T | \mathbf{p}]$ truly at random, and then we set the last column to our secret short vector $\mathbf{s}$. Of course, including $\mathbf{s}$ in the public key would reveal the key used for decryption. So, we mask $\mathbf{s}$ with a random lattice point, and set $\mathbf{p} = \mathbf{A}_0^T \mathbf{w} + \mathbf{s}$, where $\mathbf{w}$ is chosen at random. Geometrically, instead of revealing $\mathbf{s}$, we reveal its coset $\mathbf{s} + \Lambda_q(\mathbf{A}_0)$, from which recovering $\mathbf{s}$ (or an equally short vector) is a computationally hard task. Clearly, the lattice $\Lambda_q([\mathbf{A}_0^T, \mathbf{p}]^T)$ still contains $\mathbf{s}$, which can be used for decryption.

Encrypting one bit at a time is not very practical, but a much more efficient system can be obtained along similar lines by embedding not just one, but many short vectors, as first proposed in [71], for a syntactically different formulation of the encryption scheme. The geometric description presented in these notes can be instantiated in several ways, depending on the length of $\mathbf{s}$. If $\mathbf{s}$ is long enough, then $\mathbf{p}$ will be distributed almost uniformly at random, and $\mathbf{A}$ will follow the distribution $\Lambda_q(n, k)$. If $\mathbf{s}$ is shorter, then the distribution of $\mathbf{A}$ will be far from $\Lambda_q(n, k)$, but it will be indistinguishable from it because $\mathbf{p}$ is pseudo-random. Either setting of parameters is reasonable, as long as the length of the vector $\mathbf{x}$ used during the encryption process is chosen accordingly. If $\mathbf{s}$ is longer, then $\mathbf{x}$ should be shorter, while if $\mathbf{s}$ is shorter, then $\mathbf{x}$ can be longer. The corresponding proof of security will invoke a statistical or computational argument when analyzing either the public key or encrypted ciphertext.

These two instantiations were originally presented as two different cryptosystems in [73] and [34], where they are described using quite different matrix formulation. Looking at these two cryptosystems through the lens of geometry and duality, [59] showed that they are in fact the same cryptosystem instantiated for different values of the parameters. [59] also suggested that the efficiency of the cryptosystem could be improved using intermediate values of the parameters and using a computational argument both during the analysis of the public key and encrypted ciphertext. Significant efficiency improvements were later demonstrated in [44] which proposed and analyzed a concrete instantiation of the general scheme of [59]. This is the currently most efficient example of a lattice based cryptosystem based on general lattices.

4.5 Identity Based Encryption

We have seen how generating a random lattice together with a single short vector is enough to build a public key encryption scheme. But some applications do require a full basis. For example, this is the case for the identity based encryption (in the random oracle model) of [34]. We recall that an identity based encryption is an encryption scheme were public keys are simply given by the parties' names or public identifiers. The corresponding secret keys are computed by a trusted entity using a master secret key. In the scheme of [34] $\mathbf{A}$ is generated together with a full trapdoor basis $\mathbf{S}$, which is used as a master key. Then the (identity based) public key encryption scheme is built as described in the previous subsection, with one twist: instead of choosing the short decryption key $\mathbf{s}$ at random, and embedding it in a new independently generated lattice, the short vector $\mathbf{s}$ is extracted (sampled) from the lattice $\Lambda_q(\mathbf{A})$ using the master key $\mathbf{S}$ to solve an instance of the ADD problem. More specifically, given a target point $\mathbf{t}$ that represents a party's identity, a corresponding secret key is computed using $\mathbf{S}$ to find a lattice point not too far from $\mathbf{t}$.

The problem of generating a random lattice together with a full short basis is studied in [5, 8]. A new, much simpler and more efficient method to generate such lattices has been recently discovered in [62].

Lattice based identity based encryption has been further studied in [22, 1, 2] yielding both solutions in the standard model, and solution to the more general *hierarchical* identity based encryption scheme, where identities for a hierarchy, and each node can delegate limited decryption capabilities to its children.

5 Conclusion

Lattice cryptography is still a very young and fast developing area, and still it has produced a substantial body of work. We conclude these notes by providing some pointers to the literature that can be used to get a deeper understanding of the concepts touched upon in this tutorial. Other surveys presenting additional perspectives on lattice cryptography include [58, 64, 74, 75]. Work in lattice cryptography can be roughly categorized into two groups: foundational work aimed at establishing that $f_{\mathbf{A}}$ is a one way function, and cryptographic applications that use the properties of $f_{\mathbf{A}}$ to solve more complex cryptographic tasks.

We start with the first category. Works in this area tend to be mathematically more involved, but that's where the real "magic" happen: connecting the average-case and worst-case complexity of lattice problems. The reader interested in getting a feeling of how the proof of Theorem 6 works, is referred to the survey [58]. From there, the reader may proceed to [63, 73]. Both papers make use of harmonic analysis and high dimensional Fourier transform, which is very useful both to yield stronger and more concise proofs. Still on the foundational side, is a line of research initiated in [57] aimed at substantially improving the efficiency of lattice cryptography, while maintaining a form of provable security, based on the use of lattices with special algebraic structure. We expect these lattices will play a major role in bringing lattice cryptography closer to practice. The original paper [57] is still a good starting point to learn about these lattices. For more recent developments in this direction see [48, 69, 68, 51, 53, 61].

On the application side, there is a huge variety of cryptographic problems that have been recently solved using lattices. But in the end, all solutions ultimately rely on the results and techniques developed in a small number of foundational papers. In fact, many applications can be described and analyzed simply using as a starting point the one-wayness of the function $f_{\mathbf{A}}$, without even any explicit reference to lattices. But we believe that the geometric perspective advocated in this notes can be of great value to understand those constructions, and interpreting the variety of solutions in the literature in terms of a small number of basic geometric concepts. For the reader interested in applications, here we give some pointers to relevant papers, organized by category. For public key encryption see [73, 72, 52, 81, 67, 40, 44]. For digital signatures see [34, 49, 47, 18]. For group and ring signatures see [35, 77, 24]. For identity based cryptography see [2, 22, 1, 78]. For (fully) homomorphic encryption see [33, 28, 80, 65, 29, 3, 82, 79, 32, 25, 19, 31, 30]. For zero-knowledge proofs and identification protocols see [23, 46, 76, 83, 41, 70]. For still more cryptographic primitives and protocols see [66, 14, 13, 17, 12, 39, 9, 71, 16].

References

[1] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (h)ibe in the standard model. In H. Gilbert, editor, *Advances in Cryptology EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 553–572. Springer Berlin / Heidelberg, 2010.

[2] S. Agrawal, D. Boneh, and X. Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical ibe. In T. Rabin, editor, *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*. Springer, 2010.

[3] C. Aguilar Melchor, P. Gaborit, and Herranz. Additively homomorphic encryption with d-operand multiplications. In T. Rabin, editor, *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 138–154. Springer, 2010.

[4] M. Ajtai. The shortest vector problem in L2 is NP-hard for randomized reductions (extended abstract). In *Proceedings of STOC '98*, pages 10–19. ACM, May 1998.

[5] M. Ajtai. Generating hard instances of the short basis problem. In J. Wiedermann, P. van Emde Boas, and M. Nielsen, editors, *Proceedings of the 26th international colloquium on Automata, Languages and Programming - ICALP '87*, volume 1644 of *LNCS*, pages 1–9. Springer, July 1999.

[6] M. Ajtai. Generating hard instances of lattice problems. *Complexity of Computations and Proofs, Quaderni di Matematica*, 13:1–32, 2004. Preliminary version in STOC 1996.

[7] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of STOC '97*, pages 284–293. ACM, May 1997.

[8] J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. In *Proceedints of STACS*, pages 75–86, 2009. Invited to Theory of Computing Systems.

[9] B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *Proceedings of Crypto*, volume 5677 of *LNCS*, pages 595–618. Springer, Aug. 2009.

[10] S. Arora, L. Babai, J. Stern, and E. Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *Journal of Computer and System Sciences*, 54(2):317–331, Apr. 1997. Preliminary version in FOCS'93.

[11] W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296:625–635, 1993.

[12] B. Barak, I. Haitner, D. Hofheinz, and Y. Ishai. Bounded key-dependent message security. In *Proceedings of Eurocrypt*, LNCS. IACR, Springer, May 2010.

[13] R. Bendlin and I. Damgård. Threshold decryption and zero-knowledge proofs for lattice-based cryptosystems. In D. Micciancio, editor, *Proc. of TCC '10*, volume 5978 of *LNCS*, pages 201–218. Springer, Feb. 2010.

[14] R. Bendlin, I. Damgrd, C. Orlandi, and S. Zakarias. Semi-homomorphic encryption and multiparty computation. In K. Paterson, editor, *Advances in Cryptology EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 169–188. Springer, 2011.

[15] J. Blömer and J.-P. Seifert. On the complexity of computing short linearly independent vectors and short bases in a lattice. In *Proceedings of STOC '99*, pages 711–720. ACM, May 1999.

[16] D. Boneh and D. Freeman. Homomorphic signatures for polynomial functions. In K. Paterson, editor, *Advances in Cryptology EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 149–168. Springer Berlin / Heidelberg, 2011.

[17] D. Boneh and D. Freeman. Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. In D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, editors, *Public Key Cryptography PKC 2011*, volume 6571 of *Lecture Notes in Computer Science*, pages 1–16. Springer Berlin / Heidelberg, 2011.

[18] X. Boyen. Lattice mixing and vanishing trapdoors : A framework for fully secure short signatures and more. In *Proceedings of PKC*, LNCS. IACR, Springer, May 2010.

[19] Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from ring-lwe and security for key dependent messages. In *CRYPTO*, 2011. To appear.

[20] J.-Y. Cai and A. P. Nerurkar. An improved worst-case to average-case connection for lattice problems (extended abstract). In *Proceedings of FOCS '97*, pages 468–477. IEEE, Oct. 1997.

[21] J.-Y. Cai and A. P. Nerurkar. Approximating the SVP to within a factor $(1 + 1/dim^{\epsilon})$ is NP-hard under randomized reductions. *Journal of Computer and System Sciences*, 59(2):221–239, Oct. 1999. Preliminary version in CCC 1998.

[22] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In H. Gilbert, editor, *Advances in Cryptology EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 523–552. Springer Berlin / Heidelberg, 2010.

[23] Cayrel, Lindner, Rückert, and Silva. Improved zero-knowledge identification with lattices. In *PROVSEC*, 2010.

[24] Cayrel, Lindner, Rückert, and Silva. A lattice-based threshold ring signature scheme. In M. Abdalla and P. S. L. M. Barreto, editors, *LATINCRYPT*, volume 6212 of *Lecture Notes in Computer Science*. Springer, 2010.

[25] J.-S. Coron, A. Mandal, D. Naccache, , and M. Tibouchi. Fully-homomorphic encryption over the integers with shorter public-keys. In *CRYPTO*, 2011. To appear.

[26] I. Dinur. Approximating SVP_{∞} to within almost-polynomial factors is NP-hard. *Theoretical Computer Science*, 285(1):55–71, 2002. Preliminary version in CIAC 2000.

[27] I. Dinur, G. Kindler, R. Raz, and S. Safra. Approximating CVP to within almost-polynomial factors is NP-hard. *Combinatorica*, 23(2):205–243, 2003. Preliminary version in FOCS 1998.

[28] C. Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of STOC*, pages 169–178. ACM, 2009.

[29] C. Gentry. Toward basing fully homomorphic encryption on worst-case hardness. In T. Rabin, editor, *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*. Springer, 2010.

[30] C. Gentry. Fully homomorphic encryption without bootstrapping. Cryptology ePrint Archive, Report 2011/277, 2011.

[31] C. Gentry and S. Halevi. Fully homomorphic encryption without squashing using depth-3 arithmetic circuits. Cryptology ePrint Archive, Report 2011/279, 2011.

[32] C. Gentry and S. Halevi. Implementing gentrys fully-homomorphic encryption scheme. In K. Paterson, editor, *Advances in Cryptology EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 129–148. Springer Berlin / Heidelberg, 2011.

[33] C. Gentry, S. Halevi, and V. Vaikuntanathan. A simple bgn-type cryptosystem from lwe. In H. Gilbert, editor, *Advances in Cryptology EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 506–522. Springer Berlin / Heidelberg, 2010.

[34] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of STOC*, pages 197–206. ACM, May 2008.

[35] D. Gordon, J. Katz, and V. Vaikuntanathan. A group signature scheme from lattice assumptions. In *ASIACRYPT*, 2010.

[36] V. Guruswami, D. Micciancio, and O. Regev. The complexity of the covering radius problem. *Computational Complexity*, 14(2):90–121, jun 2005. Preliminary version in CCC 2004.

[37] I. Haviv, V. Lyubashevsky, and O. Regev. A note on the distribution of the distance from a lattice. *Discrete and Computational Geometry*, 41(1):162–176, Jan. 2009.

[38] I. Haviv and O. Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In *Proceedings of STOC*, pages 469–477. ACM, June 2007.

[39] J. Katz and V. Vaikuntanathan. Smooth projective hashing and password-based authenticated key exchange based on lattices. In *Proceedings of Asiacrypt*, LNCS. Springer, 2009.

[40] A. Kawachi, K. Tanaka, and K. Xagawa. Multi-bit cryptosystems based on lattice problems. In *Proceedings of PKC*, volume 4450 of *LNCS*, pages 315–329. Springer, mar 2007.

[41] A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently secure identification schemes and ad hoc anonymous identification schemes based on the worst-case hardness of lattice problems. In *Proceedings of Asiacrypt*, volume 5350 of *LNCS*, pages 372–389. Springer, 2008.

[42] S. Khot. Hardness of approximating the shortest vector problem in lattices. *Journal of the ACM*, 52(5):789–808, Sept. 2005. Preliminary version in FOCS 2004.

[43] S. Khot. Hardness of approximating the shortest vector problem in high Lp norms. *J. of Computer Systems Sciences*, 72(2):206–219, 2006. Preliminary version in FOCS 2003.

[44] R. Lindner and C. Peikert. Better key sizes (and attacks) for lwe-based encryption. In A. Kiayias, editor, *Topics in Cryptology CT-RSA 2011*, volume 6558 of *Lecture Notes in Computer Science*, pages 319–339. Springer, 2011.

[45] Y.-K. Liu, V. Lyubashevsky, and D. Micciancio. On bounded distance decoding for general lattices. In *Proceedings of RANDOM*, volume 4110 of *LNCS*, pages 450–461. Springer, Aug. 2006.

[46] V. Lyubashevsky. Lattice-based identification schemes secure under active attacks. In *Proceedings of PKC*, number 4939 in LNCS, pages 162–179. Springer, mar 2008.

[47] V. Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In *Proceedings of Asiacrypt*, LNCS. Springer, 2009.

[48] V. Lyubashevsky and D. Micciancio. Generalized compact knapsacks are collision resistant. In *Proceedings of ICALP*, volume 4052 of *LNCS*, pages 144–155. Springer, July 2006.

[49] V. Lyubashevsky and D. Micciancio. Asymptotically efficient lattice-based digital signatures. In *Proceedings of TCC*, volume 4948 of *LNCS*, pages 37–54. Springer, Mar. 2008.

[50] V. Lyubashevsky and D. Micciancio. On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In *Proceedings of Crypto*, volume 5677 of *LNCS*, pages 577–594. Springer, Aug. 2009.

[51] V. Lyubashevsky, D. Micciancio, C. Peikert, and A. Rosen. SWIFFT: a modest proposal for FFT hashing. In *Fast Software Encryption – Proceedings*, volume 5086 of *LNCS*, pages 54–72. Springer, Feb. 2008.

[52] V. Lyubashevsky, A. Palacio, and G. Segev. Public-key cryptographic primitives provably as secure as subset sum. In D. Micciancio, editor, *Proceedings of TCC*, volume 5978 of *LNCS*, pages 382–400. Springer, Feb. 2010.

[53] V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In H. Gilbert, editor, *Advances in Cryptology EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23. Springer Berlin / Heidelberg, 2010.

[54] D. Micciancio. Improving lattice based cryptosystems using the Hermite normal form. In *Proceedings of CaLC '01*, volume 2146 of *LNCS*, pages 126–145. Springer, Mar. 2001.

[55] D. Micciancio. The shortest vector problem is NP-hard to approximate to within some constant. *SIAM Journal on Computing*, 30(6):2008–2035, Mar. 2001. Preliminary version in FOCS 1998.

[56] D. Micciancio. Almost perfect lattices, the covering radius problem, and applications to Ajtai's connection factor. *SIAM Journal on Computing*, 34(1):118–169, 2004. Preliminary version in STOC 2002.

[57] D. Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, Dec. 2007. Preliminary version in FOCS 2002.

[58] D. Micciancio. *The LLL Algorith: Survey and Applcations*, chapter Cryptographic functions from worst-case complexity assumptions. Information Security and Cryptography. Springer, 2009.

[59] D. Micciancio. Duality in lattice cryptography. In *Proceedings of PKC*, LNCS. IACR, Springer, May 2010. Invited talk. Slides available from author's web page.

[60] D. Micciancio and S. 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, Mar. 2002.

[61] D. Micciancio and P. Mol. Pseudorandom knapsacks and the sample complexity of lwe search-to-decision reductions. In *CRYPTO*, 2011. To appear.

[62] D. Micciancio and C. Peikert. Trapdoor for lattices: Simpler, tighter, faster, smaller. Manuscript, 2011.

[63] D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measure. *SIAM Journal on Computing*, 37(1):267–302, 2007. Preliminary version in FOCS 2004.

[64] D. Micciancio and O. Regev. Lattice-based cryptography. In *Post-quantum cryptography*. Springer, 2008.

[65] N. Ogura, G. Yamamoto, T. Kobayashi, and S. Uchiyama. An improvement of key generation algorithm for gentry's homomorphic encryption scheme. In *IWSEC*, volume 6434 of *LNCS*, pages 70–83. Springer, 2010.

[66] A. O'Neill, C. Peikert, and B. Waters. Bi-deniable public-key encryption. In *CRYPTO*, 2011. To appear.

[67] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *Proceedings of STOC*, pages 333–342. ACM, 2009.

[68] C. Peikert and A. Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *Proceedings of TCC*, volume 3876 of *LNCS*, pages 145–166. Springer, Mar. 2006.

[69] C. Peikert and A. Rosen. Lattices that admit logarithmic worst-case to average-case connection factors. In *Proceedings of STOC*, pages 478–487. ACM, June 2007.

[70] C. Peikert and V. Vaikuntanathan. Noninteractive statistical zero-knowledge proofs for lattice problems. In *Proceedings of Crypto*, volume 5157 of *LNCS*, pages 536–553. Springer, Aug. 2008.

[71] C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for efficient and composable oblivious transfer. In *Proceedings of Crypto*, volume 5157 of *LNCS*, pages 554–571. Springer, Aug. 2008.

[72] C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *Proceedings of STOC*, pages 187–196. ACM, May 2008. Invited to SIAM J. Computing.

[73] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of ACM*, 56(6):34, Sept. 2009. Preliminary version in STOC 2005.

[74] O. Regev. Learning with errors over rings. In *ANTS*, 2010.

[75] O. Regev. The learning with errors problem (invited survey). In *CCC*, 2010.

[76] M. Rückert. Adaptively secure identity-based identification from lattices without random oracles. In *SCN*, 2010.

[77] M. Rückert. Lattice-based blind signatures. In *ASIACRYPT*, 2010.

[78] M. Rückert. Strongly unforgeable signatures and hierarchical identity-based signatures from lattices without random oracles. In *Post Quantum Cryptography – Proceedings*, 2010.

[79] N. Smart and F. Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In *Proceedings of PKC*, LNCS. IACR, Springer, May 2010.

[80] D. Stehlé and R. Steinfeld. Faster fully homomorphic encryption. In M. Abe, editor, *ASIACRYPT*, volume 6477 of *Lecture Notes in Computer Science*, pages 377–394. Springer, 2010.

[81] D. Stehlé, R. Steinfeld, K. Tanaka, and K. Xagawa. Efficient public key encryption based on ideal lattices. In *Proceedings of Asiacrypt*, LNCS. Springer, 2009.

[82] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In H. Gilbert, editor, *Advances in Cryptology EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 24–43. Springer Berlin / Heidelberg, 2010.

[83] K. Xagawa and K. Tanaka. Zero-knowledge protocols for ntru: Application to identification and proof of plaintext knowledge. In *Proc. of ProvSec '09*, volume 5848 of *LNCS*, pages 198–213. Springer, Nov. 2009.