Noninteractive Statistical Zero-Knowledge Proofs for Lattice Problems

Chris Peikert[*]
SRI International

Vinod Vaikuntanathan
MIT[†]

February 18, 2008

Abstract

We construct *noninteractive statistical zero-knowledge* (NISZK) proof systems for a variety of standard approximation problems on lattices, such as the shortest independent vectors problem and the complement of the shortest vector problem. Prior proof systems for lattice problems were either interactive or leaked knowledge (or both).

Our systems are the first known NISZK proofs for any cryptographically useful problem not related to integer factorization. In addition, they are proofs of knowledge, have reasonable complexity, and generally admit efficient prover algorithms (given appropriate auxiliary input). In some cases, they even imply the first known *interactive* statistical zero-knowledge proofs for certain lattice problems.

As an additional contribution, we construct an NISZK proof for a special language representing a disjunction (OR) of two or more variables. This may be useful for potential constructions of noninteractive (computational) zero knowledge proofs for NP based on lattice assumptions.

1 Introduction

A central idea in computer science is an *interactive proof system*, which allows a (possibly unbounded) prover to convince a computationally-limited verifier that a given statement is true [Bab85, GMR89, GS86]. The beautiful notion of *zero knowledge*, introduced by Goldwasser, Micali, and Rackoff [GMR89], even allows the prover to convince the verifier while revealing *nothing more than* the truth of the statement.

Many of the well-known results about zero knowledge, e.g., that NP (and even all of IP) has zero-knowledge proofs [GMW91, BGG$^+$88], refer to *computational* zero knowledge, where security holds only against a bounded cheating verifier (typically under some complexity assumption). Yet there has also been a rich line of research concerning proof[1] systems in which the zero-knowledge property is *statistical*. The advantages of such systems include security against even *unbounded* cheating verifiers, usually without any need for unproved assumptions. Much is now known about the class

[*] This material is based upon work supported by the National Science Foundation under Grants CNS-0716786 and CNS-0749931. Any opinions, findings, and conclusions or recommedations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

[†] Part of this work was performed while at SRI International.

[1] In this work, we will be concerned exclusively with *proof* systems (as opposed to *argument* systems, in which the cheating prover is computationally bounded).

SZK of problems possessing statistical zero-knowledge proofs; for example, it does not contain NP unless the polynomial-time hierarchy collapses [For87, AH91], it is closed under complement and union [Oka00], it has natural complete (promise) problems [SV03, GV99], and it is insensitive to whether the zero-knowledge condition is defined for arbitrary *malicious* verifiers, or only for *honest* ones [GSV98].

Removing interaction. Zero-knowledge proofs inherently derive their power from interaction [GO94]. In spite of this, Blum, Feldman, and Micali [BFM88] showed how to construct meaningful *noninteractive* zero-knowledge proofs (consisting of a single message from the prover to the verifier) if the parties simply have access to a uniformly random string. Furthermore, noninteractive *computational* zero-knowledge proofs exist for all of NP under plausible cryptographic assumptions [BFM88, BDMP91, FLS99, GOS06].

Just as with interactive proofs (and for the same reasons), it is also interesting to consider noninteractive proofs where the zero-knowledge condition is statistical. Compared with SZK, much less is known about the class NISZK of problems admitting such proofs. Clearly, NISZK is a (possibly proper) subset of SZK. It is also known to have complete (promise) problems [DDPY98, GSV99], but unlike SZK, it is not known whether NISZK is closed under complement or disjunction (OR).[2] Some conditional results are also known, e.g., NISZK = SZK if and only if NISZK is closed under complement [GSV99] (though it seems far from clear whether this condition is true or not).

Applying NISZK proofs. In cryptographic schemes, the benefits of NISZK proofs are manifold: they involve a minimal number of messages, they are secure under parallel and concurrent composition, and they provide a very strong level of security against unbounded cheating provers and verifiers alike, typically without relying on any complexity assumptions. However, the only *concrete* problems of cryptographic utility known to be in NISZK are all related in some way to integer factorization, i.e., variants of quadratic residuosity [BFM88, DDP94, DDP97] and the language of "quasi-safe" prime products [GMR98].[3]

Another important consideration in applying proof systems (both interactive and noninterative) is the complexity of the prover. Generally speaking, it is *not* enough simply to have a proof system; one also needs to be able to implement the prover *efficiently* given a suitable witness or auxiliary input. For interactive SZK, several proof systems for specific problems (e.g., those of [GMR89, MV03]) admit efficient provers, and Nguyen and Vadhan recently showed that *every* language in SZK ∩ NP has an efficient prover [NV06]. For *noninteractive* statistical zero knowledge, prover efficiency is not understood so well: while the systems relating to quadratic residuosity [BFM88, DDP94, DDP97] have efficient provers, the language of quasi-safe prime products [GMR98] is known to have an efficient prover only if interaction is allowed for one component of the proof.

1.1 Lattices and Proof Systems

Ever since the foundational work of Ajtai [Ajt04] on constructing hard-on-average cryptographic functions from *worst-case* assumptions relating to *lattices*, there has been significant interest in

[2] An earlier version of [DDPY98] claimed that NISZK was closed under complement and disjunction, but the claims have since been retracted.

[3] The language of graphs having trivial automorphism group is in NISZK, as are the (NISZK-complete) "image density" [DDPY98] and "entropy approximation" [GSV99] problems, but these problems do not seem to have any immediate applications to cryptographic schemes.

characterizing the complexity of lattice problems. Proof systems have provided an excellent means of making progress in this endeavor. We review some recent results below, after introducing the basic notions.

An n-dimensional lattice in $\mathbb{R}^n$ is a periodic "grid" of points consisting of all integer linear combinations of some set of linearly independent vectors $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{R}^n$, called a *basis* of the lattice. Two of the central computational problems on lattices are the *shortest vector* problem SVP and the *closest vector* problem CVP. The goal of SVP is to find a (nonzero) lattice point whose length is minimal, given an arbitrary basis for the lattice. The goal of CVP, given an arbitrary basis and some target point $\mathbf{t} \in \mathbb{R}^n$, is to find a lattice point closest to $\mathbf{t}$. Another problem, whose importance to cryptography was first highlighted in Ajtai's work [Ajt04], is the *shortest independent vectors* problem SIVP. There the goal (given a basis) is to find n linearly independent lattice vectors, the longest of which is as short as possible. All of these problems are known to be NP-complete in the worst case (in the case of SVP, under randomized reductions) [Ajt98, vEB81, BS99], so we do not expect to obtain NISZK (or even SZK) proof systems for them.

In this work, we are primarily concerned with the natural *approximation* versions of lattice problems, phrased as promise (or "gap") problems with some approximation factor $\gamma \geq 1$. For example, the goal of GapSVP_γ is to accept any basis for which the shortest nonzero lattice vector has length at most 1, and to reject those for which it has length at least γ. One typically views the approximation factor as a function $\gamma(n)$ of the dimension of the lattice; problems become easier (or at least no harder) for increasing values of γ. Known polynomial-time algorithms for lattice problems obtain approximation factors $\gamma(n)$ that are only slightly subexponential in n [LLL82, Sch87, AKS01, AKS02]. Moreover, obtaining any $\gamma(n) = \mathrm{poly}(n)$ approximation requires exponential time and space using known algorithms [AKS01, AKS02, BN07]. Therefore, lattice problems appear quite difficult to approximate to within even moderately-large factors.

Proof systems. We now review several proof systems for the above-described lattice problems and their complements. Every known system falls into one of two categories: *interactive* proofs that generally exhibit some form of statistical zero knowledge, and *noninteractive* proofs that are *not zero knowledge* (unless, of course, the associated lattice problems are trivial).

First of all, it is apparent that GapSVP_γ, GapCVP_γ, and $\mathsf{GapSIVP}_\gamma$ have trivial NP proof systems for any $\gamma \geq 1$. (E.g., for GapSVP_γ one can simply give a nonzero lattice vector of length at most 1.) Of course, the proofs clearly leak knowledge.

Goldreich and Goldwasser [GG00] initiated the study of interactive proof systems for lattice problems, showing that the complement problems $\mathsf{coGapSVP}_\gamma$ and $\mathsf{coGapCVP}_\gamma$ have AM proof systems for $\gamma(n) = O(\sqrt{n/\log n})$ factors. In other words, there are interactive proofs that *all* nonzero vectors in a given lattice are long, and that a given point in $\mathbb{R}^n$ is *far* from a given lattice.[4] Moreover, the protocols are perfect zero knowledge for *honest* verifiers. Aharonov and Regev [AR05] showed that for slightly looser $\gamma(n) = O(\sqrt{n})$ factors, the same two problems are even in NP. In other words, for such γ the interactive proofs of [GG00] can be replaced by a *noninteractive* witness, albeit one that leaks knowledge. Building upon [GG00, AR05], Guruswami, Micciancio, and Regev [GMR05] showed analogous AM and NP proof systems for $\mathsf{coGapSIVP}_\gamma$.

Micciancio and Vadhan [MV03] gave (malicious verifier) SZK proofs with *efficient provers* for GapSVP_γ and GapCVP_γ, where $\gamma(n) = O(\sqrt{n/\log n})$. To our knowledge, there is no known zero-

[4] Because GapSVP_γ and GapCVP_γ are in $\mathsf{NP} \cap \mathsf{coAM}$ for $\gamma(n) = O(\sqrt{n/\log n})$, the main conclusion of [GG00] is that these problems are *not* NP-hard, unless the polynomial-time hierarchy collapses.

knowledge proof system for the cryptographically important $\mathsf{GapSIVP}_\gamma$ problem (even with an inefficient prover), except by a reduction to $\mathsf{coGapSVP}$ using so-called "transference theorems" for lattices [Ban93]. This reduction introduces an extra n factor in the approximation, resulting in an AM proof system for fairly loose $\gamma(n) = O(n^{1.5}/\sqrt{\log n})$ factors. The exact same comments apply for the *covering radius* problem GapCRP_γ [GMR05], where the goal is to estimate the maximum distance to the lattice over all points in $\mathbb{R}^n$.

1.2 Our Results

We construct (without any assumption) *noninteractive statistical zero-knowledge* proof systems for a variety of lattice problems, for reasonably small approximation factors $\gamma(n)$. They are the first known NISZK proofs for lattice problems, and more generally, for any cryptographically interesting problem not related to integer factorization. In addition, they are proofs of knowledge, have reasonable communication and verifier complexity, and admit efficient provers. They also imply the first known *interactive* statistical zero-knowledge proofs (against honest verifiers) for certain lattice problems. Specifically, we construct the following:

- NISZK proofs (of knowledge, with efficient provers) for the $\mathsf{GapSIVP}_\gamma$ and GapCRP_γ problems, for any factor $\gamma(n) = \omega(\sqrt{n \log n})$.[5]

  In particular, this implies the first known (even interactive) SZK proof systems for these problems with approximation factors tighter than $n^{1.5}/\sqrt{\log n}$.

- An NISZK proof (of knowledge) for $\mathsf{coGapSVP}_\gamma$, for any factor $\gamma(n) \geq 20\sqrt{n}$.

  For this proof system, we are able to give an efficient prover for any $\gamma(n) = \omega(n \cdot \sqrt{\log n})$ factor, as well as an efficient *quantum* prover algorithm for slightly tighter $\gamma(n) = O(n/\sqrt{\log n})$ factors. (The prover's advice and the proof itself are still entirely classical; only the algorithm for generating the proof is quantum.)

- An NISZK proof for a special *disjunction* language of two or more $\mathsf{coGapSVP}_\gamma$ instances. This may serve as an important ingredient in potential constructions of noninteractive (computational) zero knowledge proofs for NP based on lattice assumptions.

Our proof systems have applications to lattice-based cryptographic schemes. For example, it is widely recognized that in public-key infrastructures, a user who presents her public key to a certification authority should also prove knowledge of a corresponding secret key (lest she present a key that actually belongs to some other user). A recent work of Gentry, Peikert, and Vaikuntanathan [GPV08] presented a variety of cryptographic schemes (including "hash-and-sign" signatures and identity-based encryption) in which the secret key can be any arbitrary set of short linearly independent vectors in a lattice. Our NISZK proof systems provide a reasonably efficient and statistically-secure way to prove knowledge of secret keys in these schemes. We add that the proof systems can be made even more efficient when applied to so-called "(integer) modular lattices" (such as those used in [GPV08]), by working only with integer vectors rather than high-precision real vectors. Due to space limitations, we defer a detailed discussion of these applications to the full version.

[5] Recall that a function $g(n) = \omega(f(n))$ if $g(n)$ grows faster than $c \cdot f(n)$ for every constant $c > 0$.

We also point out that our NISZK proofs immediately imply statistically-secure *zaps*, as defined by Dwork and Naor [DN00], for the same problems. Zaps have a number of applications in general, and we suspect that they may find equally important applications in lattice-based cryptography.

1.2.1 Techniques

The main conceptual tool for achieving zero knowledge in our proof systems is a lattice quantity called the *smoothing parameter*, introduced by Micciancio and Regev [MR07] (following related work of Regev [Reg04]). The smoothing parameter was introduced for the purpose of obtaining worst-case to average-case reductions for lattice problems, but more generally, it provides a way to generate an (almost-)uniform random variable related to an *arbitrary* given lattice.

In more detail, let $\Lambda \subset \mathbb{R}^n$ be a lattice, and imagine "blurring" all the points of Λ according to some Gaussian distribution. With enough blur, the discrete structure of the lattice is entirely destroyed, and the resulting picture is (almost) uniformly-spread over $\mathbb{R}^n$. Technically, this intuitive description is equivalent to choosing a noise vector $\mathbf{e}$ from a Gaussian distribution (centered at the origin) and reducing $\mathbf{e}$ modulo any basis $\mathbf{B}$ of the lattice. (The value $\mathbf{e} \bmod \mathbf{B}$ is the unique point $\mathbf{t} \in \mathcal{P}(\mathbf{B}) = \{\sum_i c_i \mathbf{b}_i \,:\, \forall\, i, c_i \in [0,1)\}$ such that $\mathbf{t} - \mathbf{e} \in \Lambda$; it can be computed efficiently given $\mathbf{e}$ and $\mathbf{B}$.) Informally, the smoothing parameter of the lattice is the amount of noise needed to obtain a nearly uniform distribution over $\mathcal{P}(\mathbf{B})$ via this process.

Overview of our proof systems. Our NISZK proofs all share a common structure regardless of the specific lattice problem in question. It is actually most instructive to start with the zero-knowledge *simulator*, and then build the prover and verifier around it. In fact, we have already described how the simulator works: given a basis $\mathbf{B}$, it simply chooses a Gaussian noise vector $\mathbf{e}'$ and computes $\mathbf{t}' = \mathbf{e}' \bmod \mathbf{B}$. The vector $\mathbf{t}' \in \mathcal{P}(\mathbf{B})$ is the simulated random "string," and $\mathbf{e}'$ is the simulated proof.[6] In the real proof system, the random string is a uniformly random $\mathbf{t} \in \mathcal{P}(\mathbf{B})$, and the prover (suppose for now that it is unbounded) samples a proof $\mathbf{e}$ from the Gaussian distribution *conditioned on* the event $\mathbf{e} = \mathbf{t} \bmod \mathbf{B}$. The verifier simply checks that indeed $\mathbf{t} - \mathbf{e} \in \Lambda$ and that $\mathbf{e}$ is "short enough."

For statistical zero knowledge, suppose that YES instances of the lattice problem have small smoothing parameter. Then the simulated random string $\mathbf{t}' = \mathbf{e}' \bmod \mathbf{B}$ is (nearly) uniform, just as in the real system; moreover, the distribution of the simulated proof $\mathbf{e}'$ conditioned on $\mathbf{t}'$ is the exactly the same as a real proof's distribution. For completeness, we use the fact (proved in [MR07]) that an $\mathbf{e}$ generated in the specified way is indeed relatively short. For soundness, we require that in NO instances, a significant fraction of random strings $\mathbf{t} \in \mathcal{P}(\mathbf{B})$ are simply too far away from the lattice to admit any short enough proof $\mathbf{e}$. (The soundness error can of course be attentuated by composing several independent proofs in parallel.)

The two competing requirements for YES and NO instances (for zero knowledge and soundness, respectively) determine the resulting approximation factor for the particular lattice problem. For GapSIVP and GapCRP, the factor is $\approx \sqrt{n}$, but in the case of coGapSVP, it turns out to be only $\approx n$ for technical reasons. To obtain tighter $O(\sqrt{n})$ factors, we change the system in the following way. The prover simply gives many independent proofs $\mathbf{e}_i$ in parallel (for independent $\mathbf{t}_i \in \mathcal{P}(\mathbf{B})$). The verifier is more stringent: instead of simply checking the *lengths* of the $\mathbf{e}_i$s, it performs an "eigenvalue

[6] An actual random *bit* string can represent $\mathbf{t}' \in \mathcal{P}(\mathbf{B}) \subset \mathbb{R}^n$ by its coefficients $c_i \in [0,1)$ relative to the basis $\mathbf{B}$, up to any desired level of precision.

test" that first appeared in the NP proof system of Aharonov and Regev [AR05]. Although the test and its purpose (soundness) are the same, we employ it in a technically different way: whereas in [AR05] it bounds a certain quantity computed by the verifier (which leaks knowledge, but guarantees rejection), here it bounds the volume of "bad" random strings that could potentially allow for false proofs.

We now turn to the issue of prover efficiency. Recall that the prover must choose a Gaussian noise vector $\mathbf{e}$ *conditioned on* the event that $\mathbf{e} = \mathbf{t} \bmod \mathbf{B}$. Such conditional distributions, called *discrete Gaussians* over lattices, have played a key role in several recent results in complexity theory and cryptography, e.g., [AR05, MR07, Reg05, Pei07]. The recent work of [GPV08] demonstrated an *efficient* sampling algorithm for discrete Gaussian that uses any linearly independent set of short lattice vectors as advice. This yields efficient provers for the tightest $\gamma(n) = \omega(\sqrt{n \log n})$ factors for GapSIVP and GapCRP, and $\gamma(n) = \omega(n \cdot \sqrt{\log n})$ factors for coGapSVP. In this work, we also present a *quantum* sampling algorithm (using different advice) that implies an efficient quantum prover for coGapSVP, for slightly tighter $\gamma(n) = O(n/\sqrt{\log n})$ factors.

We add that all of our proof systems easily generalize to arbitrary ℓ_p norms for $p \geq 2$, under essentially the same approximation factors $\gamma(n)$. The proof systems themselves actually remain exactly the same; their analysis in the ℓ_p norm relies on general results from [Pei07].

1.3 Open Questions

Recall that SZK is closed under complement and union [Oka00] and that every langauge in SZK$\cap$NP has a statistical zero-knowledge proof with an efficient prover [NV06]. Whether NISZK has any analogous properties is a difficult open problem with many potential consequences. Our work raises versions of these questions for several *specific* problems, which may help to shed some light on the general case.

We have shown that coGapSVP$_\gamma$ has NISZK proofs for some $\gamma(n) = \text{poly}(n)$ factors; does its complement GapSVP$_\gamma$ have such proofs as well? We suspect that a positive answer to this question, combined with our proofs for the special coGapSVP disjunction language, could lead to noninteractive (computational) zero knowledge proofs for all of NP under worst-case lattice assumptions. In addition, because the *closest* vector problem GapCVP and its complement coGapCVP both admit SZK proofs, it is an interesting question whether they also admit NISZK proofs. The chief technical difficulty in addressing any of these questions seems to be that a short (or close) lattice vector guarantees nothing useful about the smoothing parameter of the lattice (or its dual). Therefore it is unclear how the simulator could generate a uniformly random string together with a meaningful proof.

The factors $\gamma(n)$ for which we can demonstrate *efficient* provers are in some cases looser than those for which we know of *inefficient* provers. The gap between these factors is solely a consequence of our limited ability to sample from discrete Gaussians. Is there some succinct (possibly quantum) advice that permits efficient sampling from a discrete Gaussian with a parameter close to the smoothing parameter of the lattice (or close to the tightest known bound on the smoothing parameter)? More generally, does every language in NISZK $\cap$ NP have an NISZK proof with an efficient prover?

Finally, although we construct an NISZK proof for a language that is structurally similar to the disjunction (OR) of many coGapSVP instances, there are additional technical constraints on the language. It would be interesting to see if these constraints could be relaxed or lifted entirely.

2 Preliminaries

2.1 Notation

For any positive integer n, $[n]$ denotes the set $\{1, \ldots, n\}$. The function log always denotes the natural logarithm. We extend any function $f(\cdot)$ to a countable set A in the following way: $f(A) = \sum_{x \in A} f(x)$. A positive function $\epsilon(\cdot)$ is *negligible* in its parameter if it decreases faster than the inverse of any polynomial, i.e., if $\epsilon(n) = n^{-\omega(1)}$. The *statistical distance* between two distributions X and Y over a countable set A is $\Delta(X, Y) = \frac{1}{2} \sum_{a \in A} |\Pr[X = a] - \Pr[Y = a]|$.

Vectors are written using bold lower-case letters, e.g., $\mathbf{x}$. Matrices are written using bold capital letters, e.g., $\mathbf{X}$. The ith column vector of $\mathbf{X}$ is denoted $\mathbf{x}_i$. We often use matrix notation to denote a set of vectors, i.e., $\mathbf{S}$ also represents the set of its column vectors. We write $\text{span}(\mathbf{v}_1, \mathbf{v}_2, \ldots)$ to denote the linear space spanned by its arguments. For a set $S \subseteq \mathbb{R}^n$, $\mathbf{v} \in \mathbb{R}^n$, and $c \in \mathbb{R}$, we let $S + \mathbf{x} = \{\mathbf{y} + \mathbf{x} \ : \ \mathbf{y} \in S\}$ and $cS = \{c\mathbf{y} \ : \ \mathbf{y} \in S\}$.

The symbol $\|\cdot\|$ denotes the Euclidean norm on $\mathbb{R}^n$. We say that the norm of a set of vectors is the norm of its longest element: $\|\mathbf{X}\| = \max_i \|\mathbf{x}_i\|$. For any $\mathbf{t} \in \mathbb{R}^n$ and set $V \subseteq \mathbb{R}^n$, the distance from $\mathbf{t}$ to V is $\text{dist}(\mathbf{t}, V) = \inf_{\mathbf{v} \in V} \text{dist}(\mathbf{t}, \mathbf{v})$.

2.2 Noninteractive Proof Systems

We consider proof systems for promise problems $\Pi = (\Pi^{\text{YES}}, \Pi^{\text{NO}})$ where each instance of the problem is associated with some value of the security parameter n, and we partition the instances into sets Π_n^{YES} and Π_n^{NO} in the natural way. In general, the value of n might be different from the length of the instance; for example, the natural security parameter for lattice problems is the dimension n of the lattice, but the input basis might be represented using many morex bits. In this work, we assume for simplicity that instances of lattice problems have lengths bounded by some fixed polynomial in the dimension n, and we treat n as the natural security parameter.

Definition 2.1 (Noninteractive Proof System)**.** A pair (P, V) is called a *noninteractive proof system* for a promise problem $\Pi = (\Pi^{\text{YES}}, \Pi^{\text{NO}})$ if P is a (possibly unbounded) probabilistic algorithm, V is a *deterministic* polynomial-time algorithm, and the following conditions hold for some functions $c(n), s(n) \in [0, 1]$ and all sufficiently large n:

- *Completeness:* For every $x \in \Pi_n^{\text{YES}}$, $\Pr[V(x, r, P(x, r)) \text{ accepts}] \geq 1 - c(n)$.
- *Soundness:* For every $x \in \Pi_n^{\text{NO}}$, $\Pr[\exists\, \pi \ : \ V(x, r, \pi) \text{ accepts}] \leq s(n)$.

The probabilities are taken over the choice of the random input r and the random choices of P. The function $c(n)$ is called the completeness error, and the function $s(n)$ is called the soundness error. For nontriviality, we require $c(n) + s(n) \leq 1 - 1/\text{poly}(n)$.

The random input r is generally chosen uniformly at random from $\{0, 1\}^{p(|x|)}$ for some fixed polynomial $p(\cdot)$. For simplicity, we define our proof systems in a model whereq the random input r is chosen from an efficiently-samplable set R_x that depends on the instance x. This is without loss of generality, because given a random string $r' \in \{0, 1\}^{p(n)}$, both prover and verifier can generate $r \in R_x$ simply by running the sampling algorithm with randomness r'.

Note that our definition of soundness is *non-adaptive*, that is, the NO instance is fixed in advance of the random input r. Certain applications may require *adaptive* soundness, in which there do not exist *any* instance $x \in \Pi_n^{\text{NO}}$ and valid proof π, except with negligible probability over the choice of r. For proof systems, a simple argument shows that non-adaptive soundness implies adaptive soundness: let $B(n) = \text{poly}(n)$ be a bound on the length of any instance in Π_n^{NO}, and compose the proof system in parallel some $\text{poly}(n)$ times to achieve (non-adaptive) soundness $2^{-(n+B(n))}$. Then by a union bound over all $x \in \Pi_n^{\text{NO}}$, the resulting proof system has adaptive soundness 2^{-n}.

Definition 2.2. A noninteractive proof system (P, V) for a promise problem $\Pi = (\Pi^{\text{YES}}, \Pi^{\text{NO}})$ is *statistical zero knowledge* if there exists a probabilistic polynomial-time algorithm S (called a simulator) such that for all $x \in \Pi^{\text{YES}}$, the statistical distance between $S(x)$ and $(r, P(x, r))$ is negligible in n:

$$\Delta(\ S(x)\ ,\ (r, P(x,r))\ ) \leq \text{negl}(n).$$

The class of promise problems having noninteractive statistical zero knowledge proof systems is denoted NISZK.

2.3 Lattices

For a matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$ whose columns $\mathbf{b}_1, \ldots, \mathbf{b}_n$ are linearly independent, the n-dimensional *lattice*[7] Λ generated by the *basis* $\mathbf{B}$ is

$$\Lambda = \mathcal{L}(\mathbf{B}) = \left\{ \mathbf{Bc} = \sum\nolimits_{i\in[n]} c_i \cdot \mathbf{b}_i \ : \ \mathbf{c} \in \mathbb{Z}^n \right\}.$$

The *fundamental parallelepiped* of $\mathbf{B}$ is the half-open set $\mathcal{P}(\mathbf{B}) = \{\sum_i c_i \mathbf{b}_i \ : \ 0 \leq c_i < 1, i \in [n]\}$. For any lattice basis $\mathbf{B}$ and point $\mathbf{x} \in R^n$, there is a unique vector $\mathbf{y} \in \mathcal{P}(\mathbf{B})$ such that $\mathbf{y} - \mathbf{x} \in \mathcal{L}(\mathbf{B})$. This vector is denoted $\mathbf{y} = \mathbf{x} \bmod \mathbf{B}$, and it can be computed in polynomial time given $\mathbf{B}$ and $\mathbf{x}$.

The *dual lattice* of Λ, denoted Λ^*, is defined to be the set

$$\Lambda^* = \{\mathbf{x} \in \mathbb{R}^n \ : \ \forall\, \mathbf{v} \in \Lambda, \langle \mathbf{x}, \mathbf{v} \rangle \in \mathbb{Z}\}$$

of all vectors having integer inner product with *all* the vectors in Λ. It is routine to verify that this set is indeed a lattice, and if $\mathbf{B}$ is a basis for Λ, then $\mathbf{B}^* = (\mathbf{B}^{-1})^T$ is a basis for Λ^*. It also follows from the symmetry of the definition that $(\Lambda^*)^* = \Lambda$.

Let $\mathcal{C}_n = \{\mathbf{x} \in \mathbb{R}^n \ : \ \|\mathbf{x}\| \leq 1\}$ be the closed unit ball. The *minimum distance* of a lattice Λ, denoted $\lambda_1(\Lambda)$, is the length of its shortest nonzero element:

$$\lambda_1(\Lambda) = \min_{0\neq\mathbf{x}\in\Lambda} \|\mathbf{x}\|.$$

More generally, the *ith successive minimum* $\lambda_i(\Lambda)$ is the smallest radius r such that the closed ball $r\mathcal{C}_n$ contains i linearly independent vectors in Λ:

$$\lambda_i(\Lambda) = \min\left\{r \in \mathbb{R} : \dim \text{span}(\Lambda \cap r\mathcal{C}_n) \geq i\right\}.$$

The *covering radius* of Λ, denoted $\mu(\Lambda)$, is the smallest radius r such that closed balls $r\mathcal{C}_n$ centered at every point of Λ cover all of $\mathbb{R}^n$:

$$\mu(\Lambda) = \max_{\mathbf{x}\in\mathbb{R}^n} \text{dist}(\mathbf{x}, \Lambda).$$

[7] Technically, this is the definition of a *full-rank* lattice, which is all we will be concerned with in this work.

2.3.1 Basic Facts

The various quantities above can be related to each other via so-called *transference theorems*, such as the following (non-trivial) result of Banaszczyk.

Lemma 2.3 ([Ban93]). *For any n-dimensional lattice Λ, we have*

$$1 \leq 2 \cdot \lambda_1(\Lambda) \cdot \mu(\Lambda^*) \leq n.$$

The next lemma shows that the covering radius μ is bounded from below by $\lambda_n/2$. We repeat the short proof for self-containment.

Lemma 2.4 ([MG02, Theorem 7.9]). *For any n-dimensional lattice Λ, we have*

$$\lambda_n(\Lambda) \leq 2\mu(\Lambda).$$

Proof. Write $\lambda_n = \lambda_n(\Lambda)$ and $\mu = \mu(\Lambda)$. Suppose for contradition that $\lambda_n > 2\mu$, and let $\epsilon > 0$ be such that $\epsilon < \lambda_n - 2\mu$. We iteratively construct a set of linearly independent lattice vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \Lambda$ as follows. For $i = 1, \ldots, n$, let $\mathbf{t}_i \in \mathbb{R}^n$ be an arbitrary vector of length $\|\mathbf{t}_i\| = \mu + \epsilon$ that is orthogonal to $\mathbf{v}_1, \ldots, \mathbf{v}_{i-1}$. Let $\mathbf{v}_i \in \Lambda$ be a lattice vector within distance μ of $\mathbf{t}_i$. Then $\mathbf{v}_i$ is linearly independent from $\mathbf{v}_1, \ldots, \mathbf{v}_{i-1}$, because $\operatorname{dist}(\mathbf{v}_i, \operatorname{span}(\mathbf{v}_1, \ldots, \mathbf{v}_{i-1})) \geq \|\mathbf{t}_i\| - \|\mathbf{v}_i - \mathbf{t}_i\| \geq \epsilon$. Moreover, $\|\mathbf{v}_i\| \leq \|\mathbf{t}_i\| + \|\mathbf{v}_i - \mathbf{t}_i\| \leq 2\mu + \epsilon < \lambda_n$, by the triangle inequality. By induction, we obtain a set of linearly independent lattice vectors of length $\|\mathbf{v}_i\| < \lambda_n$, thus contradicting the definition of λ_n. □

The next lemma establishes that a random point in $\mathcal{P}(\mathbf{B})$ is unlikely to be "close" to the lattice, where the notion of closeness is relative to the covering radius.

Lemma 2.5 ([GMR05, Lemma 4.1]). *For any lattice $\Lambda = \mathcal{L}(\mathbf{B})$,*

$$\Pr_{\mathbf{t} \in \mathcal{P}(\mathbf{B})} \left[\operatorname{dist}(\mathbf{t}, \Lambda) < \frac{\mu(\Lambda)}{2} \right] \leq \frac{1}{2},$$

where the probability is taken over $\mathbf{t} \in \mathcal{P}(\mathbf{B})$ chosen uniformly at ranodm.

2.3.2 Problems on Lattices

Here we define some standard approximation problems on lattices. We define promise (or "gap") problems $\Pi = (\Pi^{\text{YES}}, \Pi^{\text{NO}})$, where the goal is to decide whether the instance belongs to the set Π^{YES} or the set Π^{NO} (these two sets are disjoint, but not necessarily exhaustive; when the input belongs to neither set, any output is acceptable). In the complement of a promise problem, Π^{YES} and Π^{NO} are merely swapped.

Definition 2.6 (Shortest Vector Problem). An input to GapSVP_γ is a basis $\mathbf{B}$ of an n-dimensional lattice. It is a YES instance if $\lambda_1(\mathcal{L}(\mathbf{B})) \leq 1$, and is a NO instance if $\lambda_1(\mathcal{L}(\mathbf{B})) > \gamma(n)$.

Definition 2.7 (Covering Radius Problem). An input to GapCRP_γ is a basis $\mathbf{B}$ of an n-dimensional lattice. It is a YES instance if $\mu(\mathcal{L}(\mathbf{B})) \leq 1$, and is a NO instance if $\mu(\mathcal{L}(\mathbf{B})) > \gamma(n)$.

Definition 2.8 (Shortest Independent Vectors Problem). An input to $\mathsf{GapSIVP}_\gamma$ is a basis $\mathbf{B}$ of an n-dimensional lattice. It is a YES instance if $\lambda_n(\mathcal{L}(\mathbf{B})) \leq 1$, and is a NO instance if $\lambda_n(\mathcal{L}(\mathbf{B})) > \gamma$.

Note that the choice of the quantities 1 and γ in the above problems is arbitrary; by scaling the input instance, they can be replaced by β and $\beta \cdot \gamma$ (respectively) for any $\beta > 0$ without changing the problem.

2.3.3 Gaussians on Lattices

Our review of Gaussian measures over lattices follows the development by prior works [Reg04, AR05, MR07]. For any $s > 0$ define the Gaussian function centered at $\mathbf{c}$ with parameter s as:

$$\forall \mathbf{x} \in \mathbb{R}^n,\ \rho_{s,\mathbf{c}}(\mathbf{x}) = e^{-\pi \|\mathbf{x}-\mathbf{c}\|^2/s^2}.$$

The subscripts s and $\mathbf{c}$ are taken to be 1 and $\mathbf{0}$ (respectively) when omitted. The total measure associated to $\rho_{s,\mathbf{c}}$ is $\int_{\mathbf{x}\in\mathbb{R}^n} \rho_{s,\mathbf{c}}(\mathbf{x})\, d\mathbf{x} = s^n$, so we can define a continuous Gaussian distribution centered at $\mathbf{c}$ with parameter s by its probability density function

$$\forall \mathbf{x} \in \mathbb{R}^n,\ D_{s,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{s,\mathbf{c}}(\mathbf{x})}{s^n}.$$

It is possible to sample from $D_{s,\mathbf{c}}$ efficiently to within any desired level of precision. For simplicity, we use real numbers in this work and assume that we can sample from $D_{s,\mathbf{c}}$ exactly; all the arguments can be made rigorous by using a suitable degree of precision.

For any $\mathbf{c} \in \mathbb{R}^n$, real $s > 0$, and lattice Λ, define the *discrete Gaussian distribution over* Λ as:

$$\forall \mathbf{x} \in \Lambda,\ D_{\Lambda,s,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{s,\mathbf{c}}(\mathbf{x})}{\rho_{s,\mathbf{c}}(\Lambda)}.$$

(As above, we may omit the parameters s or $\mathbf{c}$.) Note that the denominator in the above expression is always finite (see, e.g., [AR05, Claim 2.4]), so the probability distribution is well-defined. Intuitively, $D_{\Lambda,s,\mathbf{c}}$ can be viewed as a "conditional" distribution, resulting from sampling $\mathbf{x} \in \mathbb{R}^n$ from a Gaussian centered at $\mathbf{c}$ with parameter s, and conditioning on the event $\mathbf{x} \in \Lambda$.

The smoothing parameter. Micciancio and Regev [MR07] introduced a lattice quantity called the *smoothing parameter*.

Definition 2.9. For an n-dimensional lattice Λ and positive real $\epsilon > 0$, the *smoothing parameter* $\eta_\epsilon(\Lambda)$ is defined to be the smallest s such that $\rho_{1/s}(\Lambda^* \backslash \{\mathbf{0}\}) \leq \epsilon$.

The name "smoothing parameter" is due to the following (informally stated) fact: if a lattice Λ is "blurred" by adding Gaussian noise with parameter $s \geq \eta_\epsilon(\Lambda)$ for some small ϵ, the resulting distribution is close to uniform over the entire space. This is made formal in the following lemma.

Lemma 2.10 ([MR07, Lemma 4.1])**.** *For any lattice $\mathcal{L}(\mathbf{B})$, $\epsilon > 0$, $s \geq \eta_\epsilon(\mathcal{L}(\mathbf{B}))$, and $\mathbf{c} \in \mathbb{R}^n$, the statistical distance between $(D_{s,\mathbf{c}} \bmod \mathbf{B})$ and the uniform distribution over $\mathcal{P}(\mathbf{B})$ is at most $\epsilon/2$.*

The smoothing parameter of an n-dimensional lattice Λ is related to other fundamental lattice quantities, such as the dual minimum distance $\lambda_1(\Lambda^*)$ and the nth successive minimum $\lambda_n(\Lambda)$.

Lemma 2.11 ([MR07, Lemma 3.2])**.** *Let Λ be any n-dimensional lattice, and let $\epsilon(n) = 2^{-n}$. Then*

$$\eta_\epsilon(\Lambda) \quad \leq \quad \sqrt{n}/\lambda_1(\Lambda^*).$$

Lemma 2.12 ([MR07, Lemma 3.3])**.** *For any n-dimensional lattice Λ and $\epsilon > 0$, we have*

$$\eta_\epsilon(\Lambda) \quad \leq \quad \lambda_n(\Lambda) \cdot \sqrt{\log(2n(1+1/\epsilon))/\pi}.$$

In particular, for any $\omega(\sqrt{\log n})$ function, there is a negligible function $\epsilon(n)$ for which

$$\eta_\epsilon(\Lambda) \quad \leq \quad \lambda_n(\Lambda) \cdot \omega(\sqrt{\log n}).$$

The smoothing parameter also influences the behavior of *discrete* Gaussian distributions over the lattice. When $s \geq \eta_\epsilon(\Lambda)$, the distribution $D_{\Lambda,s,.\mathbf{c}}$ has a number of nice properties: it is highly concentrated within a radius $s\sqrt{n}$ around its center $\mathbf{c}$, it is not concentrated too heavily in any single direction, and it is not concentrated too heavily on any fixed hyperplane. The next lemma states all these facts precisely.

Lemma 2.13 ([MR07, Lemmas 4.4 and 4.2] and [Reg05])**.** *For any n-dimensional lattice Λ, any $\mathbf{c} \in \mathbb{R}^n$, and any $\epsilon \in (0,1)$ and $s \geq \eta_\epsilon(\Lambda)$, we have*

$$\Pr_{\mathbf{x} \sim D_{\Lambda,s,\mathbf{c}}}[\|\mathbf{x} - \mathbf{c}\| > s\sqrt{n}] \leq \tfrac{1+\epsilon}{1-\epsilon} \cdot 2^{-n}.$$

In addition, for any unit vector $\mathbf{u} \in \mathbb{R}^n$, we have

$$\mathop{\mathrm{E}}_{\mathbf{x} \sim D_{\Lambda,s,\mathbf{c}}}[\langle \mathbf{x} - \mathbf{c}, \mathbf{u}\rangle^2] \leq s^2 \cdot \left(\tfrac{1}{2\pi} + \tfrac{\epsilon}{1-\epsilon}\right).$$

In addition, if $s \geq \sqrt{2} \cdot \eta_\epsilon(\Lambda)$ and $H \subset \mathbb{R}^n$ is any fixed $(n-1)$-dimensional hyperplane, we have

$$\Pr_{\mathbf{x} \sim D_{\Lambda,s,\mathbf{c}}}[\mathbf{x} \in H] < \tfrac{1+\epsilon}{\sqrt{2}(1-\epsilon)}$$

3 Noninteractive Statistical Zero-Knowledge

In this section we give NISZK proof systems for a variety of standard lattice problems, as well as for a special kind of disjunction (OR) of two or more coGapSVP instances.

3.1 NISZK for Smooth-Or-Separated Problem

Here we introduce an intermediate lattice problem (actually, a family of problems parameterized by a function $\epsilon(n)$) called SOS, which stands for "smooth-or-separated." The SOS problem exactly captures the two properties we need for our first basic NISZK proof system: in YES instances, the lattice can be completely smoothed by a Gaussian with parameter 1, and in NO instances, a random point is at least $\sqrt{n}$ away from the lattice with good probability. Moreover, the SOS problem is at least as expressive as several standard lattice problems of interest, by which we mean that there are simple (deterministic) reductions to SOS from $\mathsf{GapSIVP}_\gamma$, GapCRP_γ, and $\mathsf{coGapSVP}_\gamma$ (for appropriate approximation factors γ).

Definition 3.1 (Smooth-Or-Separated Problem)**.** For any positive function $\epsilon = \epsilon(n)$, an input to ϵ-SOS_γ is a basis $\mathbf{B}$ of an n-dimensional lattice. It is a YES instance if $\eta_\epsilon(\mathcal{L}(\mathbf{B})) \leq 1$, and is a NO instance if $\mu(\mathcal{L}(\mathbf{B})) > \gamma(n)$.[8]

The NISZK proof system for SOS is described precisely in Figure 1. For the moment, we ignore issues of efficiency and assume that the prover is unbounded (in Section 3.3 below, we describe efficient provers for specific problems of interest). To summarize, the random input is a uniformly random point $\mathbf{t} \in \mathcal{P}(\mathbf{B})$, where $\mathbf{B}$ is the input basis. The prover samples a vector $\mathbf{e}$ from a Gaussian (centered at the origin), *conditioned* on the event that $\mathbf{e}$ is congruent to $\mathbf{t}$ modulo the lattice, i.e.,

[8] Using techniques from, e.g., [MR07], it is straightforward to verify that the YES and NO sets are disjoint whenever $\gamma \geq \sqrt{n}$ and $\epsilon(n) \leq 1/2$.

$\mathbf{e} - \mathbf{t} \in \mathcal{L}(\mathbf{B})$. (In other words, the prover samples from a *discrete* Gaussian distribution.) The verifier checks that $\mathbf{e}$ and $\mathbf{t}$ are indeed congruent, and that $\|\mathbf{e}\| \leq \sqrt{n}$.

In the YES case, the smoothing parameter is at most 1. This lets us prove that the sampled error vector $\mathbf{e}$ is indeed shorter than $\sqrt{n}$ (with overwhelming probability), ensuring completeness. More interestingly, it means that the simulator can first choose $\mathbf{e}$ from a *continuous* Gaussian, and then set the random input $\mathbf{t} = \mathbf{e} \bmod \mathbf{B}$. This $\mathbf{t}$ will be almost-uniform in $\mathcal{P}(\mathbf{B})$, ensuring zero knowledge. In the NO case, the covering radius of the lattice is large. Therefore, with good probability the random vector $\mathbf{t} \in \mathcal{P}(\mathbf{B})$ is simply too far away from the lattice to admit any short enough $\mathbf{e}$, hence no proof will convince the verifier.

NISZK proof system for SOS

Common Input: A basis $\mathbf{B}$ of an n-dimensional lattice $\Lambda = \mathcal{L}(\mathbf{B})$.

Random Input: A vector $\mathbf{t} \in \mathbb{R}^n$ chosen uniformly at random from $\mathcal{P}(\mathbf{B})$.

Prover P: Sample $\mathbf{v} \sim D_{\Lambda,-\mathbf{t}}$, and output $\mathbf{e} = \mathbf{t} + \mathbf{v} \in \mathbb{R}^n$ as the proof.

Verifier V: Accept if $\mathbf{e} - \mathbf{t} \in \Lambda$ and if $\|\mathbf{e}\| \leq \sqrt{n}$, otherwise reject.

Figure 1: The noninteractive zero-knowledge proof system for the SOS problem.

Theorem 3.2. *For any $\gamma(n) \geq 2\sqrt{n}$ and any negligible function $\epsilon(n)$, the problem ϵ-$\mathsf{SOS}_\gamma \in \mathsf{NISZK}$ via the proof system described in Figure 1.*

Furthermore, the system is a proof of knowledge in the following sense. There is a strict polynomial-time knowledge extractor E which, given an arbitrary basis $\mathbf{B}$ and oracle access to any (possibly malicious) prover P^* that satisfies the verifier on $\mathbf{B}$ with probability strictly greater than $1/2$ (over the choice of the random input and P^*'s randomness), outputs n linearly independent vectors in $\mathcal{L}(\mathbf{B})$ having length at most $9\sqrt{n \log n}$ with probability $1 - 2^{-\Omega(n)}$.

Proof. We analyze the proof system and demonstrate each of the required properties.

Completeness. Suppose that $\mathbf{B}$ is a YES instance of ϵ-SOS_γ, i.e., $\eta_\epsilon(\Lambda) \leq 1$. By construction, $\mathbf{v} \in \Lambda$ because the support of $D_{\Lambda,-\mathbf{t}}$ is Λ. Therefore $\mathbf{e} - \mathbf{t} = \mathbf{v} \in \Lambda$. Furthermore, by Lemma 2.13, we have
$$\|\mathbf{e}\| = \|\mathbf{v} - (-\mathbf{t})\| \leq \sqrt{n},$$
except with probability at most $\frac{1+\epsilon}{1-\epsilon} \cdot 2^{-n} \leq 2^{-n+1}$, which is therefore an upper bound on the completeness error.

Soundness. Suppose that $\mathbf{B}$ is a NO instance of ϵ-SOS_γ, i.e., $\mu(\Lambda) > \gamma(n) \geq 2\sqrt{n}$. For the verifier to accept, it must be that $\|\mathbf{e}\| \leq \sqrt{n}$ and $\mathbf{t} - \mathbf{e} \in \Lambda$, which implies
$$\mathrm{dist}(\mathbf{t}, \Lambda) \leq \sqrt{n} < \frac{\mu(\Lambda)}{2}.$$
However, by Lemma 2.5, the probability (over the choice of $\mathbf{t} \in \mathcal{P}(\mathbf{B})$) of this event is at most $1/2$, which is therefore an upper bound on the soundness error.

Statistical zero knowledge. The simulator S algorithm does the following: choose $\mathbf{e}' \sim D$ from the continuous Gaussian distribution (centered at $\mathbf{0}$ with parameter 1) and compute $\mathbf{t}' = \mathbf{e}' \bmod \mathcal{P}(\mathbf{B})$. Output $(\mathbf{t}', \mathbf{e}')$.

We now analyze the statistical distance between the distribution of the simulator's output $(\mathbf{t}', \mathbf{e}')$ and the distribution of $(\mathbf{t}, \mathbf{e})$ in the real protocol. By definition, the distribution of $\mathbf{v} = \mathbf{e} - \mathbf{t}$ conditioned on any fixed value of $\mathbf{t} \in \mathcal{P}(\mathbf{B})$ is exactly $D_{\Lambda,-\mathbf{t}}$. In the simulation, consider the distribution D of $\mathbf{v}' = \mathbf{e}' - \mathbf{t}'$ conditioned on any fixed value of $\mathbf{t}' \in \mathcal{P}(\mathbf{B})$. The support of D is Λ, and the distribution is

$$D(\mathbf{v}') = \frac{\rho(\mathbf{e}')}{\rho(\Lambda + \mathbf{t}')} = \frac{\rho_{-\mathbf{t}'}(\mathbf{v}')}{\rho_{-\mathbf{t}'}(\Lambda)} = D_{\Lambda,-\mathbf{t}'}(\mathbf{v}').$$

Therefore for any $\hat{\mathbf{t}} \in \mathcal{P}(\mathbf{B})$, the distribution of the simulated proof $\mathbf{e}'$ conditioned on $\mathbf{t}' = \hat{\mathbf{t}}$ is *identical* to the distribution of the real proof $\mathbf{e}$ conditioned on $\mathbf{t} = \hat{\mathbf{t}}$.

Now because $\mathbf{B}$ is a YES instance of ϵ-SOS_γ, we have $\eta_\epsilon(\Lambda) \leq 1$. By Lemma 2.10, the random input $\mathbf{t}'$ generated by the simulator is within negligible statistical distance $\epsilon(n)/2$ of uniform over $\mathcal{P}(\mathbf{B})$, which is the distribution of $\mathbf{t}$ in the real system. It follows that $\Delta((\mathbf{e}, \mathbf{t}), (\mathbf{e}', \mathbf{t}')) \leq \epsilon(n)/2$, as desired.

Proof of knowledge. Let $\mathbf{B}$ be *any* basis (not necessarily a YES instance of SOS) of an n-dimensional lattice $\Lambda = \mathcal{L}(\mathbf{B})$, and suppose that for some (possibly malicious) prover P^*,

$$\Pr_{\mathbf{t}}[V(\mathbf{B}, \mathbf{t}, P^*(\mathbf{B}, \mathbf{t})) \text{ accepts}] > 1/2,$$

where the probability is taken over a uniformly random $\mathbf{t} \in \mathcal{P}(\mathbf{B})$.

By the soundness argument above, it must be the case that $\mu(\Lambda) \leq 2\sqrt{n}$. By Lemma 2.4 and Lemma 2.12, we have

$$\eta_\epsilon(\Lambda) \leq \lambda_n(\Lambda) \cdot \sqrt{\log n} \leq 4\sqrt{n \log n}$$

for $\epsilon = 1/10$ and all sufficiently large n.

The knowledge extractor E, on input $\mathbf{B}$ and given oracle access to P^*, works as follows. Repeat the following n^2 times: choose $\mathbf{e}' \sim D_s$ where $s = 8\sqrt{n \log n}$ and compute $\mathbf{t} = \mathbf{e}' \bmod \mathcal{P}(\mathbf{B})$. Run the prover P^* on $(\mathbf{B}, \mathbf{t})$. If the prover produces an $\mathbf{e}$ such that $\|\mathbf{e}\| \leq \sqrt{n}$ and $\mathbf{e} \equiv \mathbf{t} \bmod \mathcal{P}(\mathbf{B})$ (i.e., an $\mathbf{e}$ such that $V(\mathbf{B}, \mathbf{t}, \mathbf{e})$ accepts), store the vector $\mathbf{e}' - \mathbf{e} \in \Lambda$. When finished, look among the stored vectors for a linearly independent set, and output that set if found.

First, note that the random input $\mathbf{t}$ produced by the extractor is within statistical distance $1/20$ of uniform, by Lemma 2.10 and the choice of ϵ above. Thus, P^* produces an accepting proof with probability at least $\frac{1}{4}$ over the extractor's choice of $\mathbf{t}$. Moreover, whenever P^* produces an accepting proof $\mathbf{e}$ for some fixed value of $\mathbf{t}$, it can be seen that the conditional distribution of $\mathbf{e}' - \mathbf{e}$ is exactly the discrete Gaussian $D_{\Lambda,s,-\mathbf{e}}$. By the fact that $\|\mathbf{e}\| \leq \sqrt{n}$, the triangle inequality, and Lemma 2.13, we have

$$\left\|\mathbf{e}' - \mathbf{e}\right\| \leq \left\|(\mathbf{e}' - \mathbf{e}) - (-\mathbf{e})\right\| + \|\mathbf{e}\| \leq (s+1)\sqrt{n} \leq 9\sqrt{n \log n},$$

except with probability at most 2^{-n}.

Finally, we show that n^2 iterations suffice to obtain n linearly independent vectors. When the prover produces an accepting proof $\mathbf{e}$, the probability that $\mathbf{e}' - \mathbf{e}$ lands in any fixed subspace of $\mathbb{R}^n$ is at most $9/10$ by Lemma 2.13. By a standard repetition argument, the extractor fails to find n linearly independent vectors with at most $2^{-\Omega(n)}$ probability.

This completes the proof of Theorem 3.2. □

Note that the soundness error of the basic proof system for SOS is only $1/2$ (the completeness error is exponentially small, the simulation error is at most $\epsilon(n)$, and the knowledge error is $1/2$). Of course, the soundness and knowledge errors can be attenuated to any negligible function $\delta(n)$ simply by repeating the proof system independently and in parallel $\lg(1/\delta(n)) = \omega(\log n)$ times.

3.2 Standard Lattice Problems

We now show simple, deterministic reductions from the standard lattice problems $\mathsf{GapSIVP}_{\gamma'}$, $\mathsf{GapCRP}_{\gamma'}$ and $\mathsf{coGapSVP}_{\gamma'}$ to the $\epsilon\text{-}\mathsf{SOS}_\gamma$ problem, for appropriate approximation factors $\gamma'(n)$ related to $\gamma(n)$ (and some negligible function $\epsilon(n)$).

Theorem 3.3. *For every $\gamma(n) \geq 1$ and any fixed $\omega(\sqrt{\log n})$ function, there is a deterministic polynomial-time reduction from each of the following problems to $\epsilon\text{-}\mathsf{SOS}_\gamma$ (for some negligible function $\epsilon(n)$):*

- $\mathsf{GapSIVP}_{\gamma'}$ *for any $\gamma'(n) \geq \omega(\sqrt{\log n}) \cdot 2\gamma(n)$,*
- $\mathsf{GapCRP}_{\gamma'}$ *for any $\gamma'(n) \geq \omega(\sqrt{\log n}) \cdot 2\gamma(n)$,*
- $\mathsf{coGapSVP}_{\gamma'}$ *for any $\gamma'(n) \geq 2\sqrt{n} \cdot \gamma(n)$.*

In particular, the problems $\mathsf{GapSIVP}_{\omega(\sqrt{n \log n})}$, $\mathsf{GapCRP}_{\omega(\sqrt{n \log n})}$, and $\mathsf{coGapSVP}_{4n}$ are in NISZK.

Proof. The "in particular" part of the claim follows by Theorem 3.2, which gives an NISZK proof system for $\epsilon\text{-}\mathsf{SOS}_\gamma$ for any $\gamma(n) \geq 2\sqrt{n}$.

First consider $\mathsf{GapSIVP}_{\gamma'}$. The reduction is the trivial one that on input $\mathbf{B}$ outputs $\mathbf{B}$. Without loss of generality, we can assume (by scaling) that for YES instances $\mathbf{B}$ of $\mathsf{GapSIVP}_{\gamma'}$, the lattice $\Lambda = \mathcal{L}(\mathbf{B})$ has $\lambda_n(\Lambda) \leq 1/\omega(\sqrt{\log n})$, while NO instances are such that $\lambda_n(\Lambda) > 2\gamma(n)$. Suppose that $\mathbf{B}$ is a YES instance. By Lemma 2.12, there is a negligible function $\epsilon(n)$ such that $\eta_\epsilon(\Lambda) \leq 1$, and thus $\mathbf{B}$ is a YES instance of $\epsilon\text{-}\mathsf{SOS}_\gamma$. Now suppose that $\mathbf{B}$ is a NO instance. Then by Lemma 2.4, $\mu(\Lambda) \geq \lambda_n(\Lambda)/2 > \gamma(n)$, and $\mathbf{B}$ is a NO instances of $\epsilon\text{-}\mathsf{SOS}_\gamma$.

The reduction from $\mathsf{GapCRP}_{\gamma'}$ is likewise the trivial reduction. Without loss of generality, we can assume that for YES instances $\mathbf{B}$ of $\mathsf{GapCRP}_{\gamma'}$, the lattice $\Lambda = \mathcal{L}(\mathbf{B})$ has $\mu(\Lambda) \leq 1/(2\omega(\sqrt{\log n}))$, while NO instances are such that $\mu(\Lambda) > \gamma(n)$. For a YES instance, by Lemma 2.4 and 2.12, there is a negligible function $\epsilon(n)$ such that $\eta_\epsilon(\Lambda) \leq 1$, and thus $\mathbf{B}$ is a YES instance of $\epsilon\text{-}\mathsf{SOS}_\gamma$. Now if $\mathbf{B}$ is a NO instance of $\mathsf{GapCRP}_{\gamma'}$, then we have already seen that $\mu(\Lambda) > \gamma(n)$, and $\mathbf{B}$ is a NO instances of $\epsilon\text{-}\mathsf{SOS}_\gamma$ as desired.

The reduction from $\mathsf{coGapSVP}_{\gamma'}$ is almost as trivial: on input basis $\mathbf{B}$ output the basis $\mathbf{B}^* = (\mathbf{B}^{-1})^T$ of the dual lattice $\Lambda^* = \mathcal{L}(\mathbf{B}^*)$. Without loss of generality, we can assume that YES instances (of $\mathsf{coGapSVP}_{\gamma'}$) are such that $\lambda_1(\Lambda) \geq \sqrt{n}$, while NO instances are such that $\lambda_1(\Lambda) < 1/(2\gamma(n))$. For a YES instance, we have $\eta_\epsilon(\Lambda^*) \leq \sqrt{n}/\lambda_1(\Lambda) \leq 1$ for $\epsilon(n) = 2^{-n}$ by Lemma 2.11, so $\mathbf{B}^*$ is a YES instance of $\epsilon\text{-}\mathsf{SOS}_\gamma$. For a NO instance, we have $\mu(\Lambda^*) \geq 1/(2\lambda_1(\Lambda)) > \gamma(n)$ by Lemma 2.3, so $\mathbf{B}^*$ is a NO instance of $\epsilon\text{-}\mathsf{SOS}_\gamma$, as desired. □

3.3 Prover Efficiency

We now show an efficient implementation of the prover strategy from the proof system of Figure 1, given some auxiliary information about the lattice $\Lambda = \mathcal{L}(\mathbf{B})$. Note that the prover has to sample from the discrete Gaussian distribution $D_{\Lambda,-\mathbf{t}}$ (with parameter 1). For this purpose, we use a recent result of Gentry, Peikert and Vaikuntanathan [GPV08]. They showed that, given an arbitrary full-rank set $\mathbf{S} \subset \Lambda$ (i.e., n linearly independent lattice vectors), it is possible to sample from $D_{\Lambda,s,\mathbf{c}}$ efficiently for any $\mathbf{c} \in \mathbb{R}^n$ and any $s \geq \|\mathbf{S}\| \cdot \omega(\sqrt{\log n})$.

Proposition 3.4 ([GPV08])**.** *There is a probabilistic polynomial-time algorithm* $\mathsf{SampleD}$ *having the following properties. On input an n-dimensional lattice basis $\mathbf{B}$, a full-rank set $\mathbf{S} \subset \mathcal{L}(\mathbf{B})$, any $s \geq \|\mathbf{S}\| \cdot \omega(\sqrt{\log n})$, and an arbitrary $\mathbf{c} \in \mathbb{R}^n$, the output distribution of* $\mathsf{SampleD}(\mathbf{B}, \mathbf{S}, s, \mathbf{c})$ *is within negligible (in n) statistical distance of $D_{\mathcal{L}(\mathbf{B}),s,\mathbf{c}}$.*

Corollary 3.5. *The problems* $\mathsf{GapSIVP}_{\omega(\sqrt{n \log n})}$, $\mathsf{GapCRP}_{\omega(\sqrt{n \log n})}$, *and* $\mathsf{coGapSVP}_{\omega(n^{1.5}\sqrt{\log n})}$ *admit* NISZK *proof systems with efficient prover algorithms.*

Proof. By the reductions from the proof of Theorem 3.3, $\mathsf{GapSIVP}_{\omega(\sqrt{n \log n})}$ and $\mathsf{GapCRP}_{\omega(\sqrt{n \log n})}$ reduce to $\epsilon\text{-}\mathsf{SOS}_{2\sqrt{n}}$ (for some negligible $\epsilon(n)$), which has an NISZK proof system as described in Figure 1. It simply remains to show that the prover algorithm from that system can be implemented by an efficient algorithm with suitable auxiliary input. As we have seen in the proof of Theorem 3.3, the YES instances $\mathbf{B}$ for both $\mathsf{GapSIVP}$ and GapCRP have $\lambda_n(\mathcal{L}(\mathbf{B})) \leq 1/\omega(\sqrt{\log n})$. Using an arbitrary full-rank set $\mathbf{S}$ such that $\|\mathbf{S}\| \leq 1/\omega(\sqrt{\log n})$ as the auxiliary input, an efficient prover can use the $\mathsf{SampleD}$ algorithm claimed in Proposition 3.4 to sample from a distribution that is statistically close to $D_{\mathcal{L}(\mathbf{B}),-\mathbf{t}}$ for any $\mathbf{t}$. Because the distributions are statistically close, the completeness, soundness, and knowledge errors of the efficient-prover system are negligibly close to those of the unbounded-prover system.

For $\mathsf{coGapSVP}_{\omega(n^{1.5}\sqrt{\log n})}$, by Lemmas 2.3 and 2.4 we have a reduction to $\mathsf{GapSIVP}_{\omega(\sqrt{n \log n})}$ that maps $\mathbf{B}$ to $\mathbf{B}^*$. The claim follows from the result we have already shown. □

3.4 Tighter Factors for coGapSVP

Theorem 3.3 and Corollary 3.5 establish that $\mathsf{GapSIVP}_\gamma$ and GapCRP_γ are in NISZK, and even have efficient prover strategies, for any $\gamma(n) = \omega(\sqrt{n \log n})$. On the other hand, for $\mathsf{coGapSVP}$ they give NISZK proof systems only for $\gamma(n) \geq 4n$; for prover efficiency, the factor $\gamma(n) = \omega(n^{1.5} \log n)$ is looser still.

Here we give a more sophisticated NISZK proof system specifically for $\mathsf{coGapSVP}_\gamma$. With an unbounded prover, the approximation factor is some $\gamma(n) = O(\sqrt{n})$; with an efficient prover it can be any $\gamma(n) = \omega(n\sqrt{\log n})$. Interestingly, we can also give an efficient *quantum* prover strategy for any $\gamma(n) = O(n/\sqrt{\log n})$ function, which is more than a $\log n$ factor tighter than the classical factor. (Note that the auxiliary input and proof system are still entirely classical; only the internal prover algorithm itself is quantum.)

Theorem 3.6. *For any $\gamma(n) \geq 20\sqrt{n}$, the problem* $\mathsf{coGapSVP}_\gamma$ *is in* NISZK*, via the proof system described in Figure 2.*

Furthermore, for any $\gamma(n) \geq \omega(n\sqrt{\log n})$, the prover can be implemented efficiently with an appropriate succinct witness. For any $\gamma(n) \geq n/\sqrt{\log n}$, the prover can be implemented efficiently as a quantum *algorithm with a succinct classical witness.*

Proof. We analyze the proof system and prove each of the required properties. Without loss of generality, we can assume (by scaling) that YES instances of $\mathsf{coGapSVP}$ are such that $\lambda_1(\Lambda) > \sqrt{n}$, while NO instances are such that $\lambda_1(\Lambda) \leq 1/20$.

NISZK proof system for $\mathsf{coGapSVP}$

Common Input: A basis $\mathbf{B}$ of an n-dimensional lattice $\Lambda = \mathcal{L}(\mathbf{B})$. Let $N = 10n^3 \log n$.

Random Input: Vectors $\mathbf{t}_1, \ldots, \mathbf{t}_N \in \mathcal{P}(\mathbf{B}^*)$ chosen independently and uniformly at random from $\mathcal{P}(\mathbf{B}^*)$, defining the matrix $\mathbf{T} \in (\mathcal{P}(\mathbf{B}^*))^N \subset \mathbb{R}^{n \times N}$.

Prover P: For each $i \in [N]$, choose $\mathbf{v}_i \sim D_{\Lambda^*, -\mathbf{t}_i}$, and let $\mathbf{e}_i = \mathbf{t}_i + \mathbf{v}_i$. The proof is the matrix $\mathbf{E} \in \mathbb{R}^{n \times N}$.

Verifier V: Accept if both of the following conditions hold, otherwise reject:

1. $\mathbf{e}_i - \mathbf{t}_i \in \Lambda^*$ for all $i \in [N]$, and
2. All the eigenvalues of the $n \times n$ positive semidefinite matrix $\mathbf{E}\mathbf{E}^T$ are at most $3N$.

Figure 2: The noninteractive zero-knowledge proof system for $\mathsf{coGapSVP}$.

Statistical zero knowledge. (This property is the easiest to demonstrate, so we dispense with it first.) The prover strategy on input $\mathbf{B}$ is exactly the prover strategy for ϵ-SOS on input $\mathbf{B}^*$ (Figure 1), run N times in parallel using the independent random $\mathbf{t}_i \in \mathcal{P}(\mathbf{B}^*)$ as random inputs. Therefore, we can simply run the simulator S from the proof of Theorem 3.2 on input $\mathbf{B}^*$ in parallel N times. It simply remains to ensure that $\mathbf{B}^*$ is a YES instance of ϵ-SOS for some negligible $\epsilon(n)$. Indeed, when $\mathbf{B}$ is a YES instance, we have $\lambda_1(\Lambda) > \sqrt{n}$, which by Lemma 2.11 implies $\eta_\epsilon(\Lambda^*) \leq 1$ for $\epsilon(n) = 2^{-n}$.

Completeness and prover efficiency. Suppose $\mathbf{B}$ is a YES instance. Because $\lambda_1(\Lambda) > \sqrt{n}$, we have $\eta_\epsilon(\Lambda^*) \leq 1$ for $\epsilon = 2^{-n}$ by Lemma 2.11. By definition of P, $\mathbf{e}_i - \mathbf{t}_i = \mathbf{v}_i \in \Lambda^*$ for all $i \in [N]$, so it remains to show that Test 2 is satisfied with significant probability. This fact is (almost) proved in [AR05, Lemma 6.2], where it is shown that all the eigenvalues of the matrix $\mathbf{E}\mathbf{E}^T$ are at most $3N$ (except with probability $2^{-\Omega(n)}$) if every column $\mathbf{e}_i$ is chosen independently according to D_{Λ^*}. In our case, the columns are distributed as $\mathbf{t} + D_{\Lambda^*, -\mathbf{t}}$, where $\eta_\epsilon(\Lambda^*) \leq 1$. The proof of [AR05, Lemma 6.2] carries through even when each column $\mathbf{e}_i$ is chosen according to a different distribution, as long as all of the distributions satisfy the two hypotheses of that lemma. Indeed, Lemma 2.13 establishes these hypotheses for the distribution $\mathbf{t} + D_{\Lambda^*, -\mathbf{t}}$. It follows that our proof system has completeness error $2^{-\Omega(n)}$, as desired.

For the efficient classical prover, we have $\lambda_1(\Lambda) > \omega(n\sqrt{\log n})$, so $\lambda_n(\Lambda^*) \leq 1/\omega(\sqrt{\log n})$ by Lemma 2.3. Therefore there is a sufficiently short full-rank set $\mathbf{S} \subset \Lambda^*$ that enable efficient sampling from $D_{\Lambda^*, \mathbf{c}}$ (Proposition 3.4). For the efficient *quantum* prover, we combine techniques from [Reg05] and [LLM06]. This involves generating a quantum state corresponding to the Fourier transform of $D_{\Lambda^*, \mathbf{c}}$ by using the algorithm of [LLM06] to decode points that are within distance $\sqrt{n} \leq \lambda_1(\Lambda) \cdot \sqrt{\log n / n}$ of Λ. As in [Reg05], we then compute the quantum Fourier transform and take a measurement, thus yielding a sample from $D_{\Lambda^*, \mathbf{c}}$. We defer the details.

Soundness. Suppose that $\mathbf{B}$ is a NO instance, i.e., $\lambda_1(\Lambda) \leq 1/20$. Consider the set of all random inputs $\mathbf{T}$ for which the verifier may be fooled into accepting, i.e.,

$$\mathsf{BAD} = \left\{ \mathbf{T} \in (\mathcal{P}(\mathbf{B}^*))^N \ : \ \exists\, \mathbf{E} \text{ such that } V(\mathbf{B}, \mathbf{T}, \mathbf{E}) \text{ accepts} \right\}.$$

We show that $\Pr_{\mathbf{T}}[\mathbf{T} \in \mathsf{BAD}] \leq 2^{-N}$, which establishes an exponentially-small soundness error.

Let $\mathbf{T} \in \mathsf{BAD}, \mathbf{E}$ be such that $V(\mathbf{B}, \mathbf{T}, \mathbf{E})$ accepts. Let $\mathbf{v}_1, \ldots, \mathbf{v}_n$ be an orthonormal eigenvector basis of $\mathbf{E}\mathbf{E}^T$ (which exists because $\mathbf{E}\mathbf{E}^T$ is positive semidefinite). Then because Test 2 is passed, for all $i \in [n]$ we have $\mathbf{E}\mathbf{E}^T\mathbf{v}_i = \kappa_i \mathbf{v}_i$ for some $0 \leq \kappa_i \leq 3N$. For any $\mathbf{x} \in \mathbb{R}^n$, we may write $\mathbf{x} = \sum_{i\in[n]} c_i \mathbf{v}_i \in \mathbb{R}^n$ for some coefficients $c_i \in \mathbb{R}$, hence $\|\mathbf{x}\|^2 = \sum_i c_i^2$. Furthermore, we have

$$\left\|\mathbf{E}^T\mathbf{x}\right\|^2 = \left\langle \mathbf{x}, (\mathbf{E}\mathbf{E}^T\mathbf{x}) \right\rangle = \left\langle \sum\nolimits_{i\in[n]} c_i \mathbf{v}_i, \sum\nolimits_{j\in[n]} \kappa_j c_j \mathbf{v}_j \right\rangle = \sum\nolimits_{i\in[n]} \kappa_i c_i^2 \leq 3N \cdot \|\mathbf{x}\|^2 .$$

In particular, if $\mathbf{x} \in \Lambda = \mathcal{L}(\mathbf{B})$ is a shortest nonzero vector in Λ, i.e., $\|\mathbf{x}\| = \lambda_1(\Lambda) \leq 1/20$, then $\left\|\mathbf{E}^T\mathbf{x}\right\| \leq \sqrt{N}/10$.

Now because Test 1 is passed, we have $\mathbf{t}_i - \mathbf{e}_i \in \Lambda^*$ for each $i \in [N]$. Because $\mathbf{x} \in \Lambda$, we have

$$\langle \mathbf{t}_i, \mathbf{x} \rangle = \langle \mathbf{t}_i - \mathbf{e}_i, \mathbf{x} \rangle + \langle \mathbf{e}_i, \mathbf{x} \rangle = \langle \mathbf{e}_i, \mathbf{x} \rangle \bmod 1.$$

Thus, $\mathbf{T}^T\mathbf{x} = \mathbf{E}^T\mathbf{x} \bmod 1$, i.e., each coordinate is congruent modulo 1. Now because $\left\|\mathbf{E}^T\mathbf{x}\right\| \in \mathcal{C}_N\sqrt{N}/10$ (recall that $\mathcal{C}_N$ is the closed unit ball in $\mathbb{R}^N$), we have established that if $\mathbf{T} \in \mathsf{BAD}$,

$$\mathbf{T}^T\mathbf{x} \in (\mathcal{C}_N\sqrt{N}/10) \bmod 1. \tag{3.1}$$

We now bound the probability that Equation (3.1) holds over the random choice of $\mathbf{T} \in (\mathcal{P}(\mathbf{B}^*))^N$. First we show that for any fixed nonzero $\mathbf{x} \in \Lambda$ (and in particular, for the $\mathbf{x}$ defined above), $\mathbf{T}^T\mathbf{x} \bmod 1$ is uniformly random in $[0,1)^N$. To see this, let $\mathbf{x} = \mathbf{B}\mathbf{z}$ for some nonzero $\mathbf{z} \in \mathbb{Z}^n$, and observe that for each $i \in [N]$, $\mathbf{t}_i = \mathbf{B}^*\mathbf{u}_i$ for uniformly random and independent $\mathbf{u}_i \in [0,1)^n$. Then

$$\langle \mathbf{t}_i, \mathbf{x} \rangle = \langle \mathbf{B}^*\mathbf{u}_i, \mathbf{B}\mathbf{z} \rangle = \mathbf{u}_i^T(\mathbf{B}^{-1}\mathbf{B})\mathbf{z} = \langle \mathbf{u}_i, \mathbf{z} \rangle ,$$

which is uniform modulo 1 because $\mathbf{z} \neq \mathbf{0}$. (Specifically, if $z_j \neq 0$ then $(\mathbf{u}_i)_j \cdot z_j \bmod 1$ is uniform and independent of the other coordinates of $\mathbf{u}$.)

Because $\mathbf{T}^T\mathbf{x} \bmod 1$ is uniform in the region $[0,1)^N$ having volume 1, $\Pr_{\mathbf{T}}[\mathbf{T} \in \mathsf{BAD}]$ is bounded from above by the volume of the region $(\mathcal{C}_N\sqrt{N}/10) \bmod 1 \subseteq [0,1)^N$, which in turn is bounded from above by

$$\mathrm{vol}(\mathcal{C}_N\sqrt{N}/10) = (\sqrt{N}/10)^N \cdot \mathrm{vol}(\mathcal{C}_N) = \frac{(\pi N/100)^{N/2}}{\Gamma(\frac{N}{2}+1)} \leq \left(\frac{\pi e N/100}{N/2}\right)^{N/2} \leq 2^{-N},$$

where the inequality follows by Stirling's approximation $\Gamma(k+1) = k! \geq (k/e)^k$. This completes the proof. □

3.5 NISZK for a Special Disjunction Language

Here we demonstrate a NISZK proof system for a special language that is structurally similar to the disjunction of many $\mathsf{coGapSVP}_\gamma$ instances. For simplicity in this section, we abuse notation identify lattices with their arbitrary bases (e.g., as problem instances or inputs to algorithms).

Definition 3.7. For a prime q, an input to $\mathsf{OR\text{-}coGapSVP}^k_{q,\gamma}$ is an n-dimensional lattice Λ such that $\lambda_1(\Lambda) > \gamma(n)$, and k superlattices $\Lambda_j \supset \Lambda$ for $j \in [k]$ such that the quotient groups Λ^*/Λ_j^* are all isomorphic to the additive group $G = \mathbb{Z}_q$.

It is a YES instance if $\lambda_1(\Lambda_i) > \gamma(n)$ for *some* $i \in [k]$, and is a NO instance if $\lambda_1(\Lambda_i) \leq 1$ for *every* $i \in [k]$.

For simplicity, we present an NISZK proof system for $\mathsf{OR\text{-}coGapSVP}^2_{q,\gamma}$ in Figure 3. The proof and its analysis generalize to any constant $k > 2$ with moderate changes (mainly, the $\sqrt{q}$ factors in the statement of Theorem 3.8 become $q^{(k-1)/k}$ factors).

NISZK proof system for $\mathsf{OR\text{-}coGapSVP}^2_{q,\gamma}$

Common Input: Lattices Λ, Λ_1, Λ_2 of dimension n as in Definition 3.7. Let $N = 10n^3 \log n$.

Random Input: Matrices $\mathbf{T}_1, \mathbf{T}_2 \in (\mathcal{P}(\Lambda^*))^N$ and group elements $\mathbf{s}_1, \ldots, \mathbf{s}_N \in G = \mathbb{Z}_q$ chosen independently and uniformly at random.

Prover P: Recall that $\lambda_1(\Lambda) > \gamma(n)$, and without loss of generality assume that $\lambda_1(\Lambda_1) > \gamma(n)$ (the other case is symmetric).

The auxiliary input to the prover is an oracle $\mathcal{O}$ (or its equivalent) that samples from $D_{\Lambda^*,\mathbf{c}}$ for any given $\mathbf{c} \in \mathbb{R}^n$, and an oracle $\mathcal{O}_1$ for sampling from $D_{\Lambda_1^*,\mathbf{c}}$.

Do the following for each $i \in [N]$ (for clarity, we omit the subscript i on all vectors):

1. Let $\mathbf{e}_2 \leftarrow \mathbf{t}_2 + D_{\Lambda^*,-\mathbf{t}_2}$, and let $\mathbf{g}_2 = (\mathbf{e}_2 - \mathbf{t}_2 \bmod \Lambda_2^*) \in (\Lambda^*/\Lambda_2^*) = G$.
2. Let $\mathbf{g}_1 = \mathbf{s} - \mathbf{g}_2 \in G = (\Lambda^*/\Lambda_1^*)$, and compute the unique $\mathbf{t}_1' \in \mathcal{P}(\Lambda_1^*)$ such that $\mathbf{t}_1' = \mathbf{t}_1 \bmod \Lambda^*$ and $(\mathbf{t}_1' - \mathbf{t}_1 \bmod \Lambda_1^*) = \mathbf{g}_1 \in (\Lambda^*/\Lambda_1^*)$.
3. Let $\mathbf{e}_1 \leftarrow \mathbf{t}_1' + D_{\Lambda_1^*,-\mathbf{t}_1'}$.

The proof consists of the matrices $\mathbf{E}_1, \mathbf{E}_2 \in \mathbb{R}^{n \times N}$ (whose N columns are the $\mathbf{e}_1$ and $\mathbf{e}_2$ vectors, respectively, constructed above for each $i \in [N]$).

Verifier V: Accept if all of the following conditions hold, otherwise reject.

- All the eigenvalues of both $\mathbf{E}_1\mathbf{E}_1^T$ and $\mathbf{E}_2\mathbf{E}_2^T$ are at most $3N$.
- For every $i \in [N]$ (again eliding the subscripts i),

$$\mathbf{e}_1 = \mathbf{t}_1 \bmod \Lambda^* \quad \text{and} \quad \mathbf{e}_2 = \mathbf{t}_2 \bmod \Lambda^* \quad \text{and} \quad \mathbf{g}_1 + \mathbf{g}_2 = \mathbf{s} \in G,$$

  where $\mathbf{g}_j = (\mathbf{e}_j - \mathbf{t}_j \bmod \Lambda_j^*) \in (\Lambda^*/\Lambda_j^*) = G$ for $j = 1, 2$.

Figure 3: The noninteractive statistical zero-knowledge proof system for the $\mathsf{OR\text{-}coGapSVP}$ problem.

Theorem 3.8. *Let $q \geq 100$ be prime and let $\gamma(n) \geq 40\sqrt{qn}$. Then the protocol in Figure 3 is a* NISZK *proof system for* $\mathsf{OR\text{-}coGapSVP}^2_{q,\gamma}$.

Furthermore, if $\gamma(n) \geq 40\sqrt{q} \cdot \omega(n\sqrt{\log n})$, then the oracles used by the prover can be implemented efficiently with appropriate succinct witnesses.

Proof sketch. By scaling, we can say that YES instances have $\lambda_1(\Lambda), \lambda_1(\Lambda_j) > \sqrt{n}$ for some $j \in \{1,2\}$, while NO instances have $\lambda_1(\Lambda_j) \leq 1/40\sqrt{q}$ for all $j \in \{1,2\}$. For the second part of the claim, we can assume that YES instances have $\lambda_1(\Lambda), \lambda_1(\Lambda_j) > \omega(n\sqrt{\log n})$.

Completeness is relatively straightforward to show. Briefly, for some $j \in \{1,2\}$ we have $\eta_\epsilon(\Lambda^*), \eta_\epsilon(\Lambda_j^*) \leq 1$ for $\epsilon = 2^{-n}$, by Lemma 2.11. For the second part of the claim, by Lemma 2.3 we have $\lambda_n(\Lambda^*), \lambda_n(\Lambda_j^*) \leq 1/\omega(\sqrt{\log n})$, so there are short full-rank sets $\mathbf{S} \subset \Lambda^*, \mathbf{S}_j \subset \Lambda_j^*$ that enable efficient sampling from discrete Gaussians over Λ^* and Λ_j^* (Proposition 3.4). As we showed in the proof of Theorem 3.6, the matrices $\mathbf{E}_j$ satisfy the eigenvalue test with overwhelming probability.

For statistical zero knowledge, the simulator does the following for each $i \in [N]$ (we elide the subscript i in the following): for $j \in \{1,2\}$, it chooses $\mathbf{e}_j \sim D$ (the continuous Gaussian with parameter 1 centered at $\mathbf{0}$) independently and computes $\mathbf{t}_j = \mathbf{e}_j \bmod \Lambda_j^*$. It then computes $\mathbf{g}_j = (\mathbf{e}_j - \mathbf{t}_j \bmod \Lambda_j^*) \in G$, and sets $\mathbf{s} = \mathbf{g}_1 + \mathbf{g}_2 \in G$. The simulated random input and proof are as in the proof system. Essentially, statistical zero knowledge follows because the $\mathbf{t}_j \in \mathcal{P}(\Lambda^*)$ are (statistically) uniform and independent. Furthermore, for YES instances, at least one of $\mathbf{g}_1, \mathbf{g}_2$ is uniformly random in G (statistically) conditioned on any fixed values of $\mathbf{t}_1, \mathbf{t}_2$.

The proof of soundness is more involved. Suppose we have a NO instance. We will show that the fraction of random inputs for which there exists some valid proof is 2^{-N}. For each $j \in \{1,2\}$, and let $\mathbf{x}_j$ be a shortest nonzero vector in Λ_j, so $\|\mathbf{x}\| \leq 1/40\sqrt{q}$. If the verifier accepts, then as in the proof of Theorem 3.6, we have

$$\|\mathbf{E}_j^T \mathbf{x}_j\| \leq \sqrt{N/400q}.$$

Moreover,

$$\mathbf{G}_j^T \mathbf{x}_j + \mathbf{T}_j^T \mathbf{x}_j = \mathbf{E}_j^T \mathbf{x}_j \bmod 1 \in \mathcal{C}_N \cdot \sqrt{N/400q} \bmod 1, \tag{3.2}$$

where the columns of $\mathbf{G}_j \in G^N$ are made up of the N group elements $\mathbf{g}_j \in G$ as in the verifier algorithm.

Define the discrete additive subgroup $H \subset [0,1)^n$ as $H = \langle \Lambda^*, \mathbf{x}_j \rangle \bmod 1$ (i.e., the inner product of every vector in Λ^* with $\mathbf{x}_j$). Because the inner product with $\mathbf{x}_j$ is a group homomorphism and $\mathbf{x}_j \in \Lambda_j$, $\mathbf{x}_j \notin \Lambda$, H is a nontrivial subgroup of $G = \mathbb{Z}_q$, hence it is isomorphic to $\mathbb{Z}_q$.

Now $\mathbf{G}_j^T \mathbf{x}_j \in H^N$, and by (3.2), it must be within radius $\sqrt{N/400q}$ (modulo 1) of $-\mathbf{T}_j^T \mathbf{x}_j$. Our goal will be to bound the number of possible values of $\mathbf{G}_j^T \mathbf{x}_j \in H^N$. Consider all the points of H^N within a radius of $\sqrt{N/400q}$ of $-\mathbf{T}_j^T \mathbf{x}_j$. If we center cubes with (axis-parallel) edges of length $1/q$ at all K such points, then by the triangle inequality, all the cubes lie within a ball of radius $\sqrt{N/400q} + \sqrt{N}/2q \leq \sqrt{N/100q}$ around $-\mathbf{T}_j^T \mathbf{x}_j$ modulo 1. Then we have

$$K \leq q^N \cdot \mathrm{vol}(\mathcal{C}_N \sqrt{N/100q}) = (Nq/100)^{N/2} \cdot \mathrm{vol}(\mathcal{C}_N) \leq (q/2)^{N/2}.$$

Now if the verifier accepts, there are at most $K^2 \leq (q/2)^N$ possible values for $(\mathbf{G}_1^T \mathbf{x}_1, \mathbf{G}_2^T \mathbf{x}_2) \in H^{2N}$. Because the homomorphism from $(\Lambda^*/\Lambda_j^*) = \mathbb{Z}_q$ to H is actually an isomorphism, we conclude that (for any fixed values of $\mathbf{T}_j$) there are at most $(q/2)^N$ possible values of the group elements $(\mathbf{s}_1, \ldots, \mathbf{s}_N) \in \mathbb{Z}_q^n$ for which the verifier may mistakenly accept. Therefore the probability that the random input lands on one of these value is at most 2^{-N}, and we are done. □

References

[AH91] William Aiello and Johan Håstad. Statistical zero-knowledge languages can be recognized in two rounds. *J. Comput. Syst. Sci.*, 42(3):327–345, 1991. Preliminary version in FOCS 1987.

[Ajt98] Miklós Ajtai. The shortest vector problem in L_2 is NP-hard for randomized reductions (extended abstract). In *STOC*, pages 10–19, 1998.

[Ajt04] Miklós Ajtai. Generating hard instances of lattice problems. *Quaderni di Matematica*, 13:1–32, 2004. Preliminary version in STOC 1996.

[AKS01] Miklós Ajtai, Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *STOC*, pages 601–610, 2001.

[AKS02] Miklós Ajtai, Ravi Kumar, and D. Sivakumar. Sampling short lattice vectors and the closest lattice vector problem. In *IEEE Conference on Computational Complexity*, pages 53–57, 2002.

[AR05] Dorit Aharonov and Oded Regev. Lattice problems in NP $\cap$ coNP. *J. ACM*, 52(5):749–765, 2005. Preliminary version in FOCS 2004.

[Bab85] László Babai. Trading group theory for randomness. In *STOC*, pages 421–429, 1985.

[Ban93] Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(4):625–635, 1993.

[BDMP91] Manuel Blum, Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Noninteractive zero-knowledge. *SIAM J. Comput.*, 20(6):1084–1118, 1991. Preliminary version in STOC 1998.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *STOC*, pages 103–112, 1988.

[BGG$^+$88] Michael Ben-Or, Oded Goldreich, Shafi Goldwasser, Johan Håstad, Joe Kilian, Silvio Micali, and Phillip Rogaway. Everything provable is provable in zero-knowledge. In *CRYPTO*, pages 37–56, 1988.

[BN07] Johannes Blömer and Stefanie Naewe. Sampling methods for shortest vectors, closest vectors and successive minima. In *ICALP*, pages 65–77, 2007.

[BS99] Johannes Blömer and Jean-Pierre Seifert. On the complexity of computing short linearly independent vectors and short bases in a lattice. In *STOC*, pages 711–720, 1999.

[DDP94] Alfredo De Santis, Giovanni Di Crescenzo, and Giuseppe Persiano. The knowledge complexity of quadratic residuosity languages. *Theor. Comput. Sci.*, 132(2):291–317, 1994.

[DDP97] Alfredo De Santis, Giovanni Di Crescenzo, and Giuseppe Persiano. Randomness-efficient non-interactive zero-knowledge (extended abstract). In *ICALP*, pages 716–726, 1997.

[DDPY98] Alfredo De Santis, Giovanni Di Crescenzo, Giuseppe Persiano, and Moti Yung. Image density is complete for non-interactive-SZK (extended abstract). In *ICALP*, pages 784–795, 1998.

[DN00] Cynthia Dwork and Moni Naor. Zaps and their applications. In *FOCS*, pages 283–293, 2000.

[FLS99] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple noninteractive zero knowledge proofs under general assumptions. *SIAM J. Comput.*, 29(1):1–28, 1999. Preliminary version in FOCS 1990.

[For87] Lance Fortnow. The complexity of perfect zero-knowledge (extended abstract). In *STOC*, pages 204–209, 1987.

[GG00] Oded Goldreich and Shafi Goldwasser. On the limits of nonapproximability of lattice problems. *J. Comput. Syst. Sci.*, 60(3):540–563, 2000. Preliminary version in STOC 1998.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989. Preliminary version in STOC 1985.

[GMR98] Rosario Gennaro, Daniele Micciancio, and Tal Rabin. An efficient non-interactive statistical zero-knowledge proof system for quasi-safe prime products. In *ACM Conference on Computer and Communications Security*, pages 67–72, 1998.

[GMR05] Venkatesan Guruswami, Daniele Micciancio, and Oded Regev. The complexity of the covering radius problem. *Computational Complexity*, 14:90–121, 2005. Preliminary version in CCC 2004.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991. Preliminary version in FOCS 1986.

[GO94] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *J. Cryptology*, 7(1):1–32, 1994.

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In *EUROCRYPT*, pages 339–358, 2006.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, 2008. To appear. Full version available at `http://eprint.iacr.org/2007/432`.

[GS86] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In *STOC*, pages 59–68, 1986.

[GSV98] Oded Goldreich, Amit Sahai, and Salil P. Vadhan. Honest-verifier statistical zero-knowledge equals general statistical zero-knowledge. In *STOC*, pages 399–408, 1998.

[GSV99] Oded Goldreich, Amit Sahai, and Salil P. Vadhan. Can statistical zero knowledge be made non-interactive? or on the relationship of SZK and NISZK. In *CRYPTO*, pages 467–484, 1999.

[GV99] Oded Goldreich and Salil P. Vadhan. Comparing entropies in statistical zero knowledge with applications to the structure of SZK. In *IEEE Conference on Computational Complexity*, pages 54–73, 1999.

[LLL82] Arjen K. Lenstra, Hendrik W. Lenstra, Jr., and Lászlό Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, December 1982.

[LLM06] Yi-Kai Liu, Vadim Lyubashevsky, and Daniele Micciancio. On bounded distance decoding for general lattices. In *APPROX-RANDOM*, pages 450–461, 2006.

[MG02] 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, 2002.

[MR07] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007. Preliminary version in FOCS 2004.

[MV03] Daniele Micciancio and Salil P. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In *CRYPTO*, pages 282–298, 2003.

[NV06] Minh-Huyen Nguyen and Salil P. Vadhan. Zero knowledge with efficient provers. In *STOC*, pages 287–295, 2006.

[Oka00] Tatsuaki Okamoto. On relationships between statistical zero-knowledge proofs. *J. Comput. Syst. Sci.*, 60(1):47–108, 2000. Preliminary version in STOC 1996.

[Pei07] Chris Peikert. Limits on the hardness of lattice problems in ℓ_p norms. In *IEEE Conference on Computational Complexity*, pages 333–346, 2007. Full version in ECCC Report TR06-148.

[Reg04] Oded Regev. New lattice-based cryptographic constructions. *J. ACM*, 51(6):899–942, 2004. Preliminary version in STOC 2003.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005. Revised version available from author's web page.

[Sch87] Claus-Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53:201–224, 1987.

[SV03] Amit Sahai and Salil P. Vadhan. A complete problem for statistical zero knowledge. *J. ACM*, 50(2):196–249, 2003. Preliminary version in FOCS 1997.

[vEB81] Peter van Emde Boas. Another NP-complete problem and the complexity of computing short vectors in a lattice. Technical Report 81-04, University of Amsterdam, 1981.