Noninteractive Statistical Zero-Knowledge Proofs for Lattice Problems

Chris Peikert[1,⋆] and Vinod Vaikuntanathan[2,⋆⋆]

[1] SRI International, cpeikert@alum.mit.edu
[2] MIT, vinodv@mit.edu

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 problems that are 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 cryptographically important lattice problems.

We also construct an NISZK proof for a special kind of disjunction (i.e., OR gate) related to the shortest vector problem. This may serve as a useful tool in 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 [7, 29, 30]. The beautiful notion of *zero knowledge*, introduced by Goldwasser, Micali, and Rackoff [29], 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 [24, 10], 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[3] systems in which the zero-knowledge property is

⋆ 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.

⋆⋆ Work performed while at SRI International. Supported in part by NSF Grant CNS-0430450.

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

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 SZK of problems possessing statistical zero-knowledge proofs; for example, it does not contain NP unless the polynomial-time hierarchy collapses [20, 2], it is closed under complement and union [38], it has natural complete (promise) problems [42, 28], and it is insensitive to whether the zero-knowledge condition is defined for arbitrary *malicious* verifiers, or only for *honest* ones [26].

Removing interaction. Zero-knowledge proofs inherently derive their power from interaction [25]. In spite of this, Blum, Feldman, and Micali [14] showed how to construct meaningful *noninteractive* zero-knowledge proofs (consisting of a single message from the prover to the verifier) if the parties simply share access to a uniformly random string. Furthermore, noninteractive *computational* zero-knowledge proofs exist for all of NP under plausible cryptographic assumptions [14, 13, 19, 31].

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 [17, 27], but unlike SZK, it is not known whether NISZK is closed under complement or disjunction (OR).[4] Some conditional results are also known, e.g., NISZK = SZK if and only if NISZK is closed under complement [27] (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 remain 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 [14–16] and the language of "quasi-safe" prime products [21].[5]

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 [29, 36]) admit efficient provers, and it was recently shown that *every* language in

[4] An earlier version of [17] claimed that NISZK was closed under complement and disjunction, but the claims have since been retracted.

[5] The language of graphs having trivial automorphism group is in NISZK, as are the (NISZK-complete) "image density" [17] and "entropy approximation" [27] problems, but these problems do not seem to have any immediate applications to cryptographic schemes.

$\mathsf{SZK} \cap \mathsf{NP}$ has an efficient prover [37]. For *noninteractive* statistical zero knowledge, prover efficiency is not understood so well: while the systems relating to quadratic residuosity [14–16] have efficient provers, the language of quasi-safe prime products [21] is known to have an efficient prover only if interaction is allowed in one component of the proof.

1.1 Lattices and Proof Systems

Ever since the foundational work of Ajtai [4] on constructing hard-on-average cryptographic functions from *worst-case* assumptions relating to *lattices*, there has been significant interest in 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 of 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 [4], is the *shortest independent vectors* problem SIVP. Here 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) [3, 44, 12], 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 [33, 43, 5, 6]. Moreover, obtaining a $\gamma(n) = \mathrm{poly}(n)$ approximation requires exponential time and space using known algorithms [5, 6, 11]. 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, or *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 [23] 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.[6] Moreover, the protocols are perfect zero knowledge for *honest* verifiers, but they are not known to have efficient provers. Aharonov and Regev [1] 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 [23] can be replaced by a *noninteractive* witness, albeit one that leaks knowledge. Building upon [23, 1], Guruswami, Micciancio, and Regev [32] showed analogous AM and NP proof systems for $\mathsf{coGapSIVP}_\gamma$.

Micciancio and Vadhan [36] 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-knowledge proof system for the cryptographically important $\mathsf{GapSIVP}_\gamma$ problem (even an interactive one), except by a reduction to $\mathsf{coGapSVP}$ using so-called "transference theorems" for lattices [8]. This reduction introduces an extra n factor in the approximation, resulting in fairly loose $\gamma(n) = O(n^{1.5}/\sqrt{\log n})$ factors. The same applies for the *covering radius* problem GapCRP [32], where the goal is to estimate the maximum distance from the lattice over all points in $\mathbb{R}^n$, and for the $\mathsf{GapGSMP}$ problem of approximating the *Gram-Schmidt minimum* of a lattice.

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 $\gamma(n)$. These are the first known NISZK proofs for lattice problems, and more generally, for any cryptographically useful 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 for certain lattice problems. Specifically, we construct the following:

- NISZK proofs (with efficient provers) for the $\mathsf{GapSIVP}_\gamma$, GapCRP_γ, and $\mathsf{GapGSMP}_\gamma$ problems, for any factor $\gamma(n) = \omega(\sqrt{n \log n})$.[7]
  In particular, this implies the first known (even interactive) SZK proof systems for these problems with approximation factors tighter than $n^{1.5}/\sqrt{\log n}$.

[6] 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 [23] is that these problems are *not* NP-hard, unless the polynomial-time hierarchy collapses.

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

– An NISZK proof for $\mathsf{coGapSVP}_\gamma$ for any factor $\gamma(n) \geq 20\sqrt{n}$. This is essentially the best we could hope for (up to constant factors) given the state of the art, because $\mathsf{coGapSVP}_\gamma$ is not even known to be in NP for any factor $\gamma(n) < \sqrt{n}$.
  For this proof system, we are able to give an efficient prover for $\gamma(n) = \omega(n \cdot \sqrt{\log n})$ factors, and an efficient *quantum* prover 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* problem of two or more $\mathsf{coGapSVP}_\gamma$ instances. As we describe in more detail below, this system may serve as an important ingredient in an eventual construction of noninteractive (computational) zero knowledge proofs for all of NP under lattice-related assumptions.

Our systems are also *proofs of knowledge* of a full-rank set of relatively "short" vectors in the given lattice. This is an important property in some of the applications to lattice-based cryptography we envision, described next.

Applications.

Public key infrastructure. 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 an "invalid" key, or one that actually belongs to some other user). A recent work of Gentry, Peikert, and Vaikuntanathan [22] constructed a variety of cryptographic schemes (including "hash-and-sign" signatures and identity-based encryption) in which the secret key can be any full-rank set of suitably "short" vectors in a public lattice. Our NISZK proof systems provide a reasonably efficient and statistically-secure way to prove knowledge of such secret keys. Implementing this idea requires some care, however, due to the exact nature of the knowledge guarantee and the fact that we are dealing with proof systems for *promise* problems.

To be more specific, a user generates a public key containing some basis $\mathbf{B}$ of a lattice Λ, and acts as the prover in the $\mathsf{GapSIVP}_\gamma$ system for (say) $\gamma \approx \sqrt{n}$. In order to satisfy the completeness hypothesis, an honest user needs to generate $\mathbf{B}$ along with a full-rank set of lattice vectors all having length at most ≈ 1. The statistical zero-knowledge condition ensures that nothing about the user's secret key is leaked to the authority. Now consider a potentially malicious user. By the soundness condition, we are guaranteed only that Λ contains a full-rank set of lattice vectors all *of length at most* γ (otherwise the user will not be able to give a convincing proof). Under this guarantee, our knowledge extractor is able to extract a full-rank set of lattice vectors of somewhat larger length $\approx \gamma \cdot \sqrt{n} \approx n$. Therefore, the extracted secret key vectors may be somewhat longer than the honestly-generated ones. Fortunately, the schemes of [22] are parameterized by a value L, so that they behave identically on any secret key consisting of vectors of length at most L. Letting L be a bound on the length of the *extracted* vectors ensures that the proof of knowledge is useful in the broader context, e.g., to a simulator that needs to generate valid signatures under the presented public key. We also remark that our NISZK proofs can be made more compact in size when applied to the hard-on-average *integer* lattices used in [22] and related works, by dealing only with integer vectors rather than high-precision real vectors.

NICZK *for all of* NP*?* Our proof systems may also be useful in constructing noninteractive *computational* zero-knowledge proof systems for all of NP based on the hardness of lattice problems. We outline a direction that follows the general approach of Blum, De Santis, Micali, and Persiano [13], who constructed an NICZK for the NP-complete language 3SAT under the quadratic residuosity assumption.

In [13], the common input is a 3SAT formula, and the auxiliary input to the prover is a satisfying assignment. The prover first chooses N, a product of two distinct primes. He associates, in a certain way, each true literal with a quadratic nonresidue from $\mathbb{Z}_N^*$, and each false literal with a quadratic residue. He proves in zero knowledge that (a) for each variable, either it or its negation is associated with a quadratic residue (thus, a variable and its negation cannot both be assigned true), and (b) for each clause, at least one of its three literals is associated with a quadratic nonresidue (thus, each clause is true under the implicit truth assignment). Thus, the entire proof involves zero-knowledge proofs of a disjunction of quadratic residuosity instances (for case (a)) and a disjunction of quadratic nonresiduosity instances (for case (b)).

We can replicate much of the above structure using lattices. Briefly, the modulus N translates to a suitably-chosen lattice Λ having *large* minimum distance, a quadratic nonresidue translates to a superlattice Λ_i of Λ also having *large* minimum distance, and a quadratic residue translates to a superlattice having *small* minimum distance. It then suffices to show in zero knowledge that (a) for each variable, the lattice associated to either it or its negation (or both) has small minimum distance, and (b) for each clause, the lattice associated to one of the variables in the clause has large minimum distance. In Section 3.2, we show how to implement part (b) by constructing an NISZK proof for a special disjunction of coGapSVP instances. However, we do not know how to prove noninteractively that one or more lattices has *small* minimum distance, i.e., a disjunction of GapSVP instances (see Section 1.3 for discussion). This seems to be the main technical barrier for obtaining NICZK for all of NP under lattice assumptions.

Finally, our NISZK proofs immediately imply statistically-secure *zaps*, as defined by Dwork and Naor [18], 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.

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 [35] (following related work of Regev [40]). 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 a 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 corresponds 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.

Our NISZK proofs all share a common structure regardless of the specific lattice problem in question. It is actually most instructive to first consider 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 common random "string," and $\mathbf{e}'$ is the simulated proof.[8] 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) generates a proof $\mathbf{e}$ by sampling 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 $\mathbf{t}$ is in the real system; moreover, the distribution of the simulated proof $\mathbf{e}'$ conditioned on $\mathbf{t}'$ is the exactly the same as the distribution of the real proof $\mathbf{e}$. For completeness, we use the fact (proved in [35]) that a real proof $\mathbf{e}$ generated in the specified way is indeed relatively short. Finally, 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 the GapSIVP, GapCRP, and GapGSMP problems, the factor is $\approx \sqrt{n}$, but for technical reasons it turns out to be only $\approx n$ for the coGapSVP problem. To obtain tighter $O(\sqrt{n})$ factors, we design a system that can be seen as a zero-knowledge analogue of the NP proof system of Aharonov and Regev [1]. Our prover simply gives many independent proofs $\mathbf{e}_i$ (as above) in parallel, for uniform and independent $\mathbf{t}_i \in \mathcal{P}(\mathbf{B})$. The verifier, rather than simply checking the *lengths* of the individual $\mathbf{e}_i$s, instead performs an "eigenvalue test" on the entire collection. Although the eigenvalue test and its purpose (soundness) are exactly the same as in [1], we use it in a techni-

[8] A random *binary string* can be used to represent a uniformly random $\mathbf{t}' \in \mathcal{P}(\mathbf{B}) \subset \mathbb{R}^n$ by its n coefficients $c_i \in [0,1)$ relative to the given basis $\mathbf{B}$, to any desired level of precision.

cally different way: whereas in [1] 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., [1, 35, 41, 39]. The recent work of [22] demonstrated an algorithm that can use any suitably "short" basis of the lattice as advice for *efficiently sampling* from a discrete Gaussian. Applying this algorithm immediately yields efficient provers for the tightest $\gamma(n) = \omega(\sqrt{n \log n})$ factors for $\mathsf{GapSIVP}$ and related problems, and $\gamma(n) = \omega(n \cdot \sqrt{\log n})$ factors for $\mathsf{coGapSVP}$. We also describe a *quantum* sampling algorithm (using different advice) that yields an efficient quantum prover for $\mathsf{coGapSVP}$, for slightly tighter $\gamma(n) = O(n/\sqrt{\log n})$ factors.

Finally, 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 ℓ_p norms relies upon general facts about discrete Gaussians due to Peikert [39].

1.3 Open Questions

Recall that SZK is closed under complement and union [38] and that every langauge in $\mathsf{SZK} \cap \mathsf{NP}$ has a statistical zero-knowledge proof with an efficient prover [37]. Whether NISZK has analogous properties is a difficult open problem with many potential consequences. Our work raises versions of these questions for *specific* problems, which may help to shed some light on the general case.

We have shown that $\mathsf{coGapSVP}_\gamma$ has NISZK proofs for certain $\gamma(n) = \mathrm{poly}(n)$ factors; does its complement GapSVP_γ have such proofs as well? As described above, we suspect that a positive answer to this question, combined with our proofs for the special $\mathsf{coGapSVP}$ disjunction problem, 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 $\mathsf{coGapCVP}$ 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 problem in $\mathsf{NISZK} \cap \mathsf{NP}$ have an NISZK proof with an efficient prover?

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

2 Preliminaries

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 $\text{dist}(\mathbf{t}, V) = \inf_{\mathbf{v} \in V} \text{dist}(\mathbf{t}, \mathbf{v})$.

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

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

The random input r is generally chosen uniformly at random from $\{0,1\}^{p(n)}$ for some fixed polynomial $p(\cdot)$. For notiational simplicity, we adopt a model in which the random input r is chosen from an efficiently-sampleable set R_x that may depend 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'.

By standard techniques, completeness and soundness errors can be reduced via parallel repetition. 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 $x \in \Pi_n^{\mathrm{NO}}$ 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 error $2^{-p(n)}$ for any desired $p(n) = \mathrm{poly}(n)$: let $B(n) = \mathrm{poly}(n)$ be a bound on the length of any instance in Π_n^{NO}, and compose the proof system in parallel some $\mathrm{poly}(n)$ times to achieve (non-adaptive) soundness $2^{-p(n)-B(n)}$. Then by a union bound over all $x \in \Pi_n^{\mathrm{NO}}$, the resulting proof system has adaptive soundness $2^{-p(n)}$.

Definition 2 (NISZK). *A noninteractive proof system (P,V) for a promise problem $\Pi = (\Pi^{YES}, \Pi^{NO})$ is* statistical zero knowledge *if there exists a probabilistic polynomial-time algorithm S (called a simulator) such that for all $x \in \Pi^{YES}$, the statistical distance between $S(x)$ and $(r, P(x,r))$ is negligible in n:*

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

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

For defining proofs of knowledge, we adapt the general approach advocated by Bellare and Goldreich [9] to our noninteractive setting. In particular, the definition is entirely distinct from that of a proof system, and it refers to *relations* (not promise problems). Let $R \subseteq \{0,1\}^* \times \{0,1\}^*$ be a binary relation where the first entry x of each $(x,y) \in R$ is associated with some value of the security parameter n, and partition the relation into sub-relations R_n in the natural way. Let $R_x = \{y \,:\, (x,y) \in R\}$ and $\Pi_n^R = \{x \,:\, \exists\, y \text{ such that } (x,y) \in R_n\}$.

Definition 3 (Noninteractive proof of knowledge). *Let R be a binary relation, let V be a determinstic polynomial time machine, and let $\kappa(n), c(n) : \mathbb{N} \to [0,1]$ be functions. We say that V is a* knowledge verifier for the relation R *with* nontriviality error c *and* knowledge error κ *if the following two conditions hold:*

1. Nontriviality (with error c)*: there exists a probabilistic function P such that for all $x \in \Pi_n^R$, $\Pr[V(x,r,P(x,r))$ accepts$] \geq 1 - c(n)$.*
2. Validity (with error κ)*: there exists a probabilistic oracle machine E such that for for every probabilistic function P^* and every $x \in \Pi_n^R$ where*

$$p_x = \Pr[V(x,r,P^*(x,r)) \text{ accepts}] > \kappa(n),$$

$E^{P^*}(x)$ outputs a string from R_x in expected time poly$(n)/(p_x - \kappa(n))$.

2.2 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*[9] Λ generated by the *basis* $\mathbf{B}$ is

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

The *fundamental parallelepiped* of $\mathbf{B}$ is the half-open set

$$\mathcal{P}(\mathbf{B}) = \{\sum_i c_i \mathbf{b}_i \ : \ 0 \le c_i < 1, i \in [n]\}.$$

For any lattice basis $\mathbf{B}$ and point $\mathbf{x} \in \mathbb{R}^n$, 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 $\mathbf{x}$.

For any (ordered) set $\mathbf{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\} \subset \mathbb{R}^n$ of linearly independent vectors, let $\tilde{\mathbf{S}} = \{\tilde{\mathbf{s}_1}, \ldots, \tilde{\mathbf{s}_n}\}$ denote its Gram-Schmidt orthogonalization, defined iteratively in the following way: $\tilde{\mathbf{s}_1} = \mathbf{s}_1$, and for each $i = 2, \ldots, n$, $\tilde{\mathbf{s}_i}$ is the component of $\mathbf{s}_i$ orthogonal to $\mathrm{span}(\mathbf{s}_1, \ldots, \mathbf{s}_{i-1})$. Clearly, $\|\tilde{\mathbf{s}_i}\| \le \|\mathbf{s}_i\|$.

Let $\mathcal{C}_n = \{\mathbf{x} \in \mathbb{R}^n \ : \ \|\mathbf{x}\| \le 1\}$ be the closed unit ball. The *minimum distance* of a lattice Λ, denoted $\lambda_1(\Lambda)$, is the length of its shortest nonzero element: $\lambda_1(\Lambda) = \min_{0 \ne \mathbf{x} \in \Lambda} \|\mathbf{x}\|$. 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: $\lambda_i(\Lambda) = \min\{r \in \mathbb{R} : \dim \mathrm{span}(\Lambda \cap r\mathcal{C}_n) \ge i\}$. The *Gram-Schmidt minimum* $\tilde{bl}(\Lambda)$ is $\tilde{bl}(\Lambda) = \min_{\mathbf{B}} \|\tilde{\mathbf{B}}\| = \min_{\mathbf{B}} \max_{i \in [n]} \|\tilde{\mathbf{b}}_i\|$, where the minimum is taken over all (ordered) bases $\mathbf{B}$ of Λ. The definition is restricted to bases without loss of generality, because for any (ordered) full-rank set $\mathbf{S} \subset \Lambda$, there is an (ordered) basis $\mathbf{B}$ of Λ such that $\|\tilde{\mathbf{B}}\| \le \|\tilde{\mathbf{S}}\|$ (see [34, Lemma 7.1]). The *covering radius* $\mu(\Lambda)$ is the smallest radius r such that closed balls $r\mathcal{C}_n$ centered at every point of Λ cover all of $\mathbb{R}^n$: $\mu(\Lambda) = \max_{\mathbf{x} \in \mathbb{R}^n} \mathrm{dist}(\mathbf{x}, \Lambda)$.

The *dual lattice* $\Lambda*$ of Λ, is 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 $(\Lambda^*)^* = \Lambda$.

Lemma 4 ([8]). *For any n-dimensional lattice Λ, $1 \le 2 \cdot \lambda_1(\Lambda) \cdot \mu(\Lambda^*) \le n$.*

Lemma 5 ([34, Theorem 7.9]). *For any n-dimensional lattice Λ,*

$$\tilde{bl}(\Lambda) \le \lambda_n(\Lambda) \le 2\mu(\Lambda).$$

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.

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

Lemma 6 ([32, Lemma 4.1]). *For any lattice $\Lambda = \mathcal{L}(\mathbf{B})$,*

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

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

We now define some standard approximation problems on lattices, all of which ask to estimate (to within some factor γ) the value of some geometric lattice quantity. We define promise (or "gap") problems $\Pi = (\Pi^{\mathrm{YES}}, \Pi^{\mathrm{NO}})$, 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 simply swapped.

Definition 7 (Lattice Problems). *Let $\gamma = \gamma(n)$ be an approximation factor in the dimension n. For any function ϕ from lattices to the positive reals, we define an approximation problem where the input is a basis $\mathbf{B}$ of an n-dimensional lattice. It is a YES instance if $\phi(\mathcal{L}(\mathbf{B})) \le 1$, and is a NO instance if $\phi(\mathcal{L}(\mathbf{B})) > \gamma(n)$.*

In particular, we define the following concrete problems by instantiating ϕ:

- *The* Shortest Vector Problem GapSVP_γ, *for $\phi = \lambda_1$.*
- *The* Shortest Independent Vectors Problem $\mathsf{GapSIVP}_\gamma$, *for $\phi = \lambda_n$.*
- *The* Gram-Schmidt Minimum Problem $\mathsf{GapGSMP}_\gamma$, *for $\phi = \tilde{bl}$.*
- *The* Covering Radius Problem GapCRP_γ, *for $\phi = \mu$.*

Note that the choice of the quantities 1 and γ above 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.

Gaussians on Lattices. Our review of Gaussian measures over lattices follows the development by prior works [40, 1, 35]. 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 $\forall \mathbf{x} \in \mathbb{R}^n,\ D_{s,\mathbf{c}}(\mathbf{x}) = \rho_{s,\mathbf{c}}(\mathbf{x})/s^n$.

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 $\mathbf{c}$.) 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 $\mathbf{x} \in \Lambda$.

Definition 8 ([35]). *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 $\epsilon > 0$, the resulting distribution is $\epsilon/2$-close to uniform over the entire space. This is made formal in the following lemma.

Lemma 9 ([35, 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 is related to other important lattice quantities.

Lemma 10 ([35, Lemma 3.2]). *Let Λ be any n-dimensional lattice, and let $\epsilon(n) = 2^{-n}$. Then $\eta_\epsilon(\Lambda) \leq \sqrt{n}/\lambda_1(\Lambda^*)$.*

Lemma 11 ([22, Lemma 3.1]). *For any n-dimensional lattice Λ and $\epsilon > 0$, we have*

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

In particular, for any $\omega(\sqrt{\log n})$ function, there is a negligible function $\epsilon(n)$ for which $\eta_\epsilon(\Lambda) \leq \tilde{bl}(\Lambda) \cdot \omega(\sqrt{\log n})$.

Note that because $\tilde{bl}(\Lambda) \leq \lambda_n(\Lambda)$, we also have $\eta_\epsilon(\Lambda) \leq \lambda_n(\Lambda) \cdot \omega(\sqrt{\log n})$; this is Lemma 3.3 in [35].

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. We refer to [35, Lemmas 4.2 and 4.4] and [41, Lemma 3.13] for precise statements of these facts.

3 Noninteractive Statistical Zero Knowledge

Here we demonstrate NISZK proofs for several natural lattice problems. Due to lack of space, we give intuitive proof sketches here and defer complete proofs to the full version.

We first 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, $\mathsf{GapGSMP}_\gamma$, and $\mathsf{coGapSVP}_\gamma$ (for appropriate approximation factors γ).

Definition 12 (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 $\mu(\mathcal{L}(\mathbf{B})) > \gamma(n)$.*[10]

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

In the YES case, the smoothing parameter is at most 1. This lets us prove that the sampled proof $\mathbf{e}$ 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 $\mathbf{t} = \mathbf{e} \bmod \mathbf{B}$. By Lemma 9, this $\mathbf{t}$ is almost-uniform in $\mathcal{P}(\mathbf{B})$, ensuring zero knowledge. In the NO case, the covering radius of the lattice is large. By Lemma 6, 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 can convince the verifier. (A complete proof of Theorem 13 below is given in the full version.)

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.

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

Theorem 13. *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. The completeness error of the system is $c(n) = 2^{-n+1}$ and the soundness error is $s(n) = 1/2$.*

By deterministic reductions to the ϵ-SOS_γ problem, several standard lattice problems are also in NISZK. The proof of the following corollary is a straightforward application of Lemmas 4, 5, 10, and 11, and is deferred to the full version.

[10] Using techniques from [35], it can be verified that the YES and NO sets are disjoint whenever $\gamma \geq \sqrt{n}$ and $\epsilon(n) \leq 1/2$.

Corollary 14. *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 ϵ-SOS_γ (for some negligible function $\epsilon(n)$):*

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

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

We now turn to the knowledge guarantee for the protocol. For a function $\epsilon = \epsilon(n)$, we define a relation R_ϵ where an instance (for security parameter n) is a basis $\mathbf{B} \subset \mathbb{R}^{n \times n}$ of a lattice having smoothing parameter η_ϵ bounded by 1 (without loss of generality), and a witness for $\mathbf{B}$ is a full-rank set $\mathbf{S} \subset \mathcal{L}(\mathbf{B})$ of lattice vectors having length at most $2\sqrt{n}$.

Theorem 15. *For any positive $\epsilon(n) \leq 1/3$, the verifier described in Figure 1 is a knowledge verifier for relation R_ϵ with nontriviality error $c(n) = 2^{-n+1}$ and knowledge error $\kappa(n) = \epsilon(n)/2$.*

Now consider the complexity of the prover in the protocol from Figure 1. 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 [22, Theorem 4.1], which shows how to sample (within negligible statistical distance) from $D_{\mathcal{L}(\mathbf{B}),s,\mathbf{t}}$ for any $s \geq \|\tilde{\mathbf{B}}\| \cdot \omega(\sqrt{\log n})$. The next corollary immediately follows (proof in the full version).

Corollary 16. *The following problems admit* NISZK *proof systems with efficient provers:* $\mathsf{GapSIVP}_{\omega(\sqrt{n \log n})}$, $\mathsf{GapCRP}_{\omega(\sqrt{n \log n})}$, *and* $\mathsf{coGapSVP}_{\omega(n^{1.5}\sqrt{\log n})}$.

3.1 Tighter Factors for coGapSVP

For $\mathsf{coGapSVP}$, Corollaries 14 and 16 give NISZK proof systems only for $\gamma(n) \geq 4n$; with an efficient prover, the factor $\gamma(n) = \omega(n^{1.5} \log n)$ is looser still.

Here we give a more sophisticated NISZK proof specifically for $\mathsf{coGapSVP}$. The proof of the next theorem is given in the full version.

Theorem 17. *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.*

NISZK proof system for 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 $3N$.

Fig. 2. The noninteractive zero-knowledge proof system for coGapSVP.

3.2 NISZK for a Special Disjunction Language

Here we consider a special language that is structurally similar to the disjunction of many $\mathsf{coGapSVP}_\gamma$ instances. For simplicity, we abuse notation and identify lattices with their arbitrary bases in problem instances.

Definition 18. *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 *$i \in [k]$.*

Theorem 19 below relates to the $\mathsf{OR\text{-}coGapSVP}^2_{q,\gamma}$ problem; it generalizes to any $k > 2$ with moderate changes (mainly, the $\sqrt{q}$ factors in the statement of Theorem 19 become $q^{(k-1)/k}$ factors).

Theorem 19. *For prime $q \geq 100$ and $\gamma(n) \geq 40\sqrt{qn}$, $\mathsf{OR\text{-}coGapSVP}^2_{q,\gamma}$ is in NISZK.*

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

References

1. Dorit Aharonov and Oded Regev. Lattice problems in NP ∩ coNP. *J. ACM*, 52(5):749–765, 2005. Preliminary version in FOCS 2004.
2. 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.
3. Miklós Ajtai. The shortest vector problem in L_2 is NP-hard for randomized reductions (extended abstract). In *STOC*, pages 10–19, 1998.
4. Miklós Ajtai. Generating hard instances of lattice problems. *Quaderni di Matematica*, 13:1–32, 2004. Preliminary version in STOC 1996.

5. Miklós Ajtai, Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *STOC*, pages 601–610, 2001.
6. 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.
7. László Babai. Trading group theory for randomness. In *STOC*, pages 421–429, 1985.
8. Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(4):625–635, 1993.
9. Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In *CRYPTO*, pages 390–420, 1992.
10. 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.
11. Johannes Blömer and Stefanie Naewe. Sampling methods for shortest vectors, closest vectors and successive minima. In *ICALP*, pages 65–77, 2007.
12. 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.
13. 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.
14. Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *STOC*, pages 103–112, 1988.
15. Alfredo De Santis, Giovanni Di Crescenzo, and Giuseppe Persiano. The knowledge complexity of quadratic residuosity languages. *Theor. Comput. Sci.*, 132(2):291–317, 1994.
16. Alfredo De Santis, Giovanni Di Crescenzo, and Giuseppe Persiano. Randomness-efficient non-interactive zero-knowledge (extended abstract). In *ICALP*, pages 716–726, 1997.
17. 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.
18. Cynthia Dwork and Moni Naor. Zaps and their applications. In *FOCS*, pages 283–293, 2000.
19. 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.
20. Lance Fortnow. The complexity of perfect zero-knowledge (extended abstract). In *STOC*, pages 204–209, 1987.
21. 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.
22. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206, 2008.
23. 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.
24. 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.

25. Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *J. Cryptology*, 7(1):1–32, 1994.
26. Oded Goldreich, Amit Sahai, and Salil P. Vadhan. Honest-verifier statistical zero-knowledge equals general statistical zero-knowledge. In *STOC*, pages 399–408, 1998.
27. 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.
28. 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.
29. 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.
30. Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In *STOC*, pages 59–68, 1986.
31. Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In *EUROCRYPT*, pages 339–358, 2006.
32. 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.
33. Arjen K. Lenstra, Hendrik W. Lenstra, Jr., and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, December 1982.
34. 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.
35. 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.
36. Daniele Micciancio and Salil P. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In *CRYPTO*, pages 282–298, 2003.
37. Minh-Huyen Nguyen and Salil P. Vadhan. Zero knowledge with efficient provers. In *STOC*, pages 287–295, 2006.
38. Tatsuaki Okamoto. On relationships between statistical zero-knowledge proofs. *J. Comput. Syst. Sci.*, 60(1):47–108, 2000. Preliminary version in STOC 1996.
39. Chris Peikert. Limits on the hardness of lattice problems in ℓ_p norms. *Computational Complexity*, 17(2):300–351, May 2008. Preliminary version in CCC 2007.
40. Oded Regev. New lattice-based cryptographic constructions. *J. ACM*, 51(6):899–942, 2004. Preliminary version in STOC 2003.
41. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005.
42. 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.
43. Claus-Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53:201–224, 1987.
44. 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.