Statistical zero-knowledge proofs with efficient provers: lattice problems and more

Daniele Micciancio[1*] and Salil Vadhan[2**]

[1] University of California, San Diego, La Jolla CA 92093, USA,
daniele@cs.ucsd.edu
[2] Harvard University, Cambridge MA 02138, USA,
salil@eecs.harvard.edu

Abstract. We construct several new statistical zero-knowledge proofs with *efficient provers*, i.e. ones where the prover strategy runs in probabilistic polynomial time given an **NP** witness for the input string.
Our first proof systems are for approximate versions of the Shortest Vector Problem (SVP) and Closest Vector Problem (CVP), where the witness is simply a short vector in the lattice or a lattice vector close to the target, respectively. Our proof systems are in fact proofs of knowledge, and as a result, we immediately obtain efficient lattice-based identification schemes which can be implemented with arbitrary families of lattices in which the approximate SVP or CVP are hard.
We then turn to the general question of whether *all* problems in **SZK** $\cap$ **NP** admit statistical zero-knowledge proofs with efficient provers. Towards this end, we give a statistical zero-knowledge proof system with an efficient prover for a natural restriction of Statistical Difference, a complete problem for **SZK**. We also suggest a plausible approach to resolving the general question in the positive.

1 Introduction

Zero-knowledge proof systems, introduced in [1], have proven to be a powerful tool for constructing cryptographic protocols. They have also turned out to be a rich object of study from the perspective of complexity theory. In this paper, we focus on *statistical* zero knowledge (**SZK**), which is the form of zero knowledge that provides the strongest security guarantees and whose complexity-theoretic study has been most active in recent years. One significant gap between much of the recent theoretical study and the cryptographic applicability of **SZK** involves the *prover's efficiency*, i.e. whether the prover can be implemented in polynomial time (given some auxiliary information). This property is clearly essential for a zero-knowledge proof to be used in cryptographic protocols, but many of the theoretical results ignore this issue. Prover efficiency for **SZK** has been considered in the past, leading to the result of Bellare and Petrank [2] that any **SZK**

* Supported in part by NSF Career Award CCR-0093029.
** Supported in part by NSF Grant CCR-0205423 and a Sloan Research Fellowship.

proof system admits a prover that runs in probabilistic polynomial time given an **NP** oracle. However, this notion of efficiency is insufficient for cryptography, as the **NP** oracle cannot be realized efficiently. In cryptographic applications, one would like the prover to run in probabilistic polynomial time given only the input string x (drawn from some **NP** language L) and an **NP**-witness w (the "secret key") that $x \in L$. We call a proof system with this property a proof system with an *efficient prover*. (These were called *prover-practical* proof systems in [3].) A number of the classic perfect and statistical zero-knowledge proof systems [1, 4] have efficient provers, but not all problems in $\mathbf{SZK} \cap \mathbf{NP}$ are known to have such proof systems. Indeed, it remains an intriguing open problem to characterize the class of problems which have statistical zero-knowledge proofs with efficient provers and extend known results about statistical zero knowledge to this class.

In this paper, we construct statistical zero-knowledge proofs with efficient provers for several problems previously not known to have such proofs. We first do this for approximate versions of the Closest Vector Problem (CVP) and Shortest Vector Problem (SVP) in lattices. These proof systems immediately yield efficient identification schemes based on the hardness of these problems. An interesting property of our schemes is that they allow us to use arbitrary lattices (where CVP and SVP are hard), which gives potential advantages both from the efficiency and security points of view; for example, there is no need to embed a "trapdoor basis" in the lattice. Then we construct a statistical zero-knowledge proof with an efficient prover for a natural restriction of Statistical Difference, which is known to be a complete problem for **SZK**. We view the latter result as progress towards characterizing the class of problems having statistical zero-knowledge proofs with efficient provers.

1.1 Statistical Zero Knowledge

Zero-knowledge proof systems are protocols by which a computationally unbounded *prover* can convince a probabilistic polynomial-time *verifier*, of an assertion, i.e. that some string x is a YES instance of some decision problem. The zero-knowledge property requires that the verifier "learns nothing" from this interaction other than the fact that the assertion being proven is true. In a *statistical* zero-knowledge *proof* system, the security for both parties is very strong. Specifically, it holds even with respect to *computationally unbounded* cheating provers or verifiers. Note that even though the security holds for computationally unbounded parties, the *prescribed* verifier strategy is always required to be polynomial time. We will discuss the prover's efficiency later. The class of problems possessing statistical zero-knowledge proofs is denoted **SZK**.

In addition to its cryptographic significance, **SZK** has turned out to be quite interesting from a complexity-theoretic perspective. On the one hand it is known to contain important computational problems, such as Graph Nonisomorphism [4] and Quadratic Residuosity [1]. On the other hand, it is contained in the class $\mathbf{AM} \cap \mathbf{co}\text{-}\mathbf{AM}$ [5, 6] and hence is unlikely to contain **NP**-hard problems. More recently, it was discovered that **SZK** is closed under complement [7] and has natural complete problems [8, 9]. Moreover, a number of useful transformations of statistical zero-knowledge proof systems have been given, for example showing that every proof system which is statistical zero knowledge for the *honest verifier* can be transformed into one which is statistical zero knowledge even for cheating verifiers [7, 10].

The above theoretical investigations focus on the traditional definition of **SZK**, whereby no computational restriction is placed on the prover strategy, and many manipulations used in the study of **SZK** do not preserve the prover's efficiency; indeed, this is inherent in the techniques used (namely, black-box transformations) [11]. Nevertheless, we consider it an important research direction to overcome this barrier and extend the study of **SZK** to protocols with efficient provers. In particular, can we characterize the subclass of **SZK** possessing statistical zero-knowledge proofs with efficient provers? Since the efficient prover property only makes sense for problems in **NP** (actually **MA**) and **SZK** is not known to be contained in **NP**,[3] so we do not hope to show that all of **SZK** has efficient provers. But do all problems in **SZK** ∩ **NP** have statistical zero-knowledge proofs with efficient provers?

1.2 Lattice Problems

A *lattice* is a subset of $\mathbb{R}^n$ consisting of all integer linear combinations of a set of linearly independent vectors. Two basic computational problems involving lattices are the Shortest Vector Problem, finding the shortest nonzero vector in the lattice, and the Closest Vector Problem, finding the lattice vector closest to a given target vector. These problems have received a great deal of attention recently in both the cryptography and complexity theory literature. On the complexity side, approximate versions of both of these problems have been shown to be **NP**-hard [15–18], and variants of the approximate Shortest Vector Problem have been shown to be related by a worst-case/average-case connection [19]. On the cryptography side, a number of cryptographic primitives have been proposed which implicitly or explicitly rely on the hardness of these problems. These include the one-way functions of [19, 20], the collision-resistant hash functions of [21, 22], the public-key encryption schemes of [23–25].

In [26], Goldreich and Goldwasser exhibited statistical zero-knowledge proofs for approximate versions of the *complements* of Shortest Vector Problem and Closest Vector Problem.[4] That is, they gave protocols for proving that a lattice has no short vector (resp., has no vector close to the target vector). The Goldreich–Goldwasser proof systems do not have efficient provers. Indeed, the problems they consider are not known to be in **NP** and their main motivation was to prove that they are in **AM** (and, being also in **co-NP**, are thus unlikely to be **NP**-hard under standard types of reductions). However, since **SZK** is closed under complement [7], it follows from their result that the corresponding approximate versions of the Shortest Vector Problem and the Closest Vector

[3] Actually, there is some recent evidence that **AM** may equal **NP** [12–14] which would imply that $\mathbf{SZK} \subseteq \mathbf{NP} \cap \mathbf{co\text{-}NP}$.

[4] In fact, their proof systems are *perfect* zero knowledge (against an honest verifier).

Problem themselves (rather than their complements) are also in **SZK**. Since these problems are in **NP**, we can hope to construct statistical zero-knowledge proofs with efficient provers for them. However, the **SZK** proofs obtained by applying the general result of [7] (or even later simplifications [8, 9, 27]) do not guarantee efficient provers, and in addition would be extremely cumbersome and impractical.

1.3 Our Results

We first construct statistical zero-knowledge proof systems with efficient provers for approximate versions of the Shortest Vector Problem and Closest Vector Problem. The approximation factor for our proof system can be as small as in the Goldreich–Goldwasser proof systems, namely $\Theta(\sqrt{n/\log n})$ where n is the rank of the lattice. The prover strategy can be implemented in polynomial time given only a short lattice vector (resp., lattice vector close to the target vector). The proof systems are actually proofs of knowledge, and hence immediately give rise to identification schemes [28] provided one can efficiently generate lattices in which either of these problems is hard together with the corresponding witnesses. We remark that in order to efficiently prove that a target point is close to the lattice (or that the lattice contains short vectors) it is not necessary to know a short (trapdoor) basis, i.e., a basis consisting entirely of short vectors. On the security side, embedding a trapdoor basis has often been regarded as a weak point for many lattice and subset-sum based cryptosystems. Our identification schemes can be instantiated with any lattice, offering the highest degree of security. For example, one can use lattices derived from the random classes of [19] or [22]. This results in provably secure lattice-based identification (ID) schemes with an average-case/worst-case connection.[5,6] On the efficiency side, complete freedom in the choice of the lattice enables the use of lattices with special structure (e.g., the cyclic lattices of [20], or the convolutional modular lattices of NTRU [25]), or share the same lattice among different users, in order to get smaller key size or faster identification procedures. (See Section 5.)

We then return to the general question of efficient provers for **SZK**. We generalize techniques of Itoh, Ohta, and Shizuya [29] to show that a natural restriction of Statistical Difference has a statistical zero-knowledge proof with an efficient (polynomial time) prover.[7] In the Statistical Difference problem, one is given two (suitably represented) probability distributions, and

[5] In order to use these lattices in our construction one needs a procedure to generate a lattice together with a short vector, but this can be achieved as explained in [19] by slightly perturbing the lattice distribution.

[6] The results of [19, 22] immediately give one way functions from worst case hardness assumptions, which, in turn, imply the existence of secure ID schemes. However, these generic constructions are pretty inefficient. Our constructions build ID schemes directly from the underlying lattice problems (i.e. without going through one-way functions), resulting in substantially more efficient ID schemes.

[7] The prover in this proof system runs in polynomial time, but is not as practical as those for the lattice problems. In particular, our results about statistical difference

the question is to determine if they are relatively close (say, within statistical distance at most $1/2$) or are far apart (say, at statistical distance at least $1-\epsilon$). This is a complete problem for **SZK** for any $0 < \epsilon < 1/\sqrt{2}$ [8]. Statistical Difference is not known to be in **NP**, so we cannot give a proof system with efficient provers for it. We consider the restriction of Statistical Difference obtained setting $\epsilon = 0$: determine if two distributions are within statistical distance $1/2$ or are completely disjoint. We observe that this problem is in **NP**, and show that it admits a statistical zero-knowledge proof system with efficient provers. Thus we view this as a step towards finding proof systems with efficient provers for all problems in **SZK** $\cap$ **NP**. In addition, the techniques we use (namely [29]) are not "black box," so this approach is not subject to the limitations in [11].

1.4 Related Work

The first zero-knowledge proof systems, namely those for Quadratic Residuosity and Quadratic Nonresiduosity [1], and Graph Isomorphism [4] had efficient provers and achieved perfect zero-knowledge. Subsequently, **SZK** proof systems with efficient provers have been found for a number of other number-theoretic problems (e.g., [3, 30]), all random self-reducible problems [31] and monotone formulae over random self-reducible problems [32]).

Other notions of prover efficiency (mostly interesting from the perspective of computational complexity) have been considered before. Building upon previous work, Bellare and Petrank [2] show that for any **SZK** proof system, it is possible to implement the prover strategy in probabilistic polynomial time given an **NP** oracle. Notice that given an **NP** oracle for Satisfiability, one can efficiently find **NP**-witnesses for arbitrary **NP** problems, by the self-reducibility of **NP**-complete problems (such as Satisfiability). So, the provers considered in [2], are considerably more powerful than ours, and allow one to prove arbitrary **SZK** languages, even those outside **NP**.

A more restrictive notion of prover efficiency is considered in [33], where the prover is given oracle access to a decision oracle for the same language L underlying the proof system.[8] For example, the (honest-verifier) perfect zero-knowledge proof system for Graph Nonisomorphism [4] satisfies this notion of prover efficiency. The results of [33] are negative: there are **NP** languages for which finding an **NP** witness for $x \in L$, or even proving membership $x \in L$ interactively (whether or not in zero-knowledge), cannot be efficiently reduced to deciding membership in L. This notion of proof system, called *competitive*

should be regarded as a plausibility result aimed at characterizing the complexity class of statistical zero-knowledge proof systems with efficient provers, rather than a concrete proposal of a proof system to be used in cryptographic applications.

[8] When L is an **NP**-complete problem, then these provers are as powerful as those of [2]. However, **SZK** is not likely to contain any **NP**-complete problem. So, for an arbitrary language L in **SZK**, it is not clear how to efficiently prove membership in L given oracle access to a decision procedure for L.

in [33], is incomparable with ours. On the one hand, our provers are given an input string x together with an **NP**-witness for $x \in L$, and it is not clear how to efficiently compute such a witness given only a decision oracle for L when L is not **NP**-complete or self-reducible. On the other hand, the provers of [33] can make queries "$y \in L$?" to the oracle for arbitrary strings y (possibly different from the input string x), while our prover is only given a witness for the input string x.

In any case, the notions of prover efficiency considered by [2, 33] and related papers, seem mostly interesting from a computational complexity perspective, and do not match the requirements of cryptographic applications. A crucial difference is that the notion we study here makes sense only for problems in **NP**, while the results of [2, 33] apply to languages outside **NP** as well.

Organization. The rest of the paper is organized as follows. In Section 2 we give some basic definitions about statistical difference and the lattice problems studied in this paper. In Section 3 we present and analyze the proof system for CVP. The proof system for SVP is sketched in Section 4. Section 5 discusses our lattice based identification schemes. Finally, in Section 6 we study STATISTICAL DIFFERENCE, and the problem of designing **SZK** proofs with efficient provers for all problems in **SZK** $\cap$ **NP**. Because of space constraints, most proofs are not presented here, and can be found in the full version of the paper.

2 Preliminaries

In this section we recall some basic definitions and techniques that will be used in the rest of the paper. For more details the reader is referred to the books [34, 35] or the papers in the references.

2.1 Statistical difference

The statistical distance between two discrete random variables X and Y over a (countable) set A is the quantity $\Delta(X,Y) = \frac{1}{2}\sum_{a \in A} |\Pr\{X = a\} - \Pr\{Y = a\}|$.

STATISTICAL DIFFERENCE is a collection of problems (parameterized by two real numbers $0 \leq \alpha < \beta \leq 1$) of the form: given two succinctly specified probability distributions, decide whether they are statistically close or statistically far apart. The probability distributions are specified by circuits which sample from them. That is, we are given a circuit $X : \{0,1\}^m \to \{0,1\}^n$ which we interpret as specifying the probability distribution $X(U_m)$ on $\{0,1\}^n$, where U_m is the uniform probability distribution over $\{0,1\}^m$. More formally, for $0 \leq \alpha < \beta \leq 1$, we define the following promise problem.

Definition 1 (STATISTICAL DIFFERENCE)**.** *Instances of promise problem* $\mathrm{SD}^{\alpha,\beta}$ *are pairs* (X,Y) *where* X *and* Y *are probability distributions.* (X,Y) *is a* YES *instance if* $\Delta(X,Y) \leq \alpha$*, and a* NO *instance if* $\Delta(X,Y) \geq \beta$*. (We have defined these problems as the complements of those defined in [8], because this formulation is more convenient for our purposes.)*

In [8] it is shown that $\mathrm{SD}^{\alpha,\beta}$ is complete for **SZK** for all $0 < \beta/2 < \alpha < \beta^2 < 1$. In particular $\mathrm{SD}^{1/3,2/3}$ is **SZK**-complete, and $\mathrm{SD}^{1/2,1-\epsilon}$ is **SZK**-complete for all $0 < \epsilon < 1/\sqrt{2}$.

2.2 Lattice problems and technical tools

Let $\mathbb{R}^m$ be the m-dimensional Euclidean space. A *lattice* in $\mathbb{R}^m$ is the set of all integral combinations of n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ in $\mathbb{R}^m$ $(m \geq n)$. The integers n and m are called the *rank* and *dimension* of the lattice, respectively. Using matrix notation, if $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$, the lattice generated by basis matrix $\mathbf{B}$ is $\mathcal{L}(\mathbf{B}) = \{\mathbf{Bx} : \mathbf{x} \in \mathbb{Z}^n\}$, where $\mathbf{Bx}$ is the usual matrix-vector multiplication. For computational purposes, $\mathbf{B}$ and $\mathbf{y}$ are usually restricted to have integer (or, equivalently, rational) entries. In this paper, we will occasionally use real vectors in order to simplify the exposition. However, the use of real numbers is not essential, and integer or rational approximations can always be substituted for real vectors whenever they occur. Moreover, we often assume that the lattice if full rank, i.e., $n = m$, as any lattice can be transformed into a full-rank *real* lattice.

Approximate versions of the Shortest Vector Problem and Closest Vector Problem described in the introduction are captured by the promise problems GapSVP_γ and GapCVP_γ defined as follows.

Definition 2. *Instances of promise problem* GapSVP_γ *are pairs* $(\mathbf{B}, t)$ *where* $\mathbf{B} \in \mathbb{Z}^{m \times n}$ *is a lattice basis and* $t \in \mathbb{Q}$ *a rational number.* $(\mathbf{B}, t)$ *is a* YES *instance if* $\|\mathbf{Bx}\| \leq t$ *for some* $\mathbf{x} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$. $(\mathbf{B}, t)$ *is a* NO *instance if* $\|\mathbf{Bx}\| > \gamma t$ *for all* $\mathbf{x} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$.

Definition 3. *Instances of promise problem* GapCVP_γ *are triples* $(\mathbf{B}, \mathbf{y}, t)$ *where* $\mathbf{B} \in \mathbb{Z}^{m \times n}$ *is a lattice basis,* $\mathbf{y} \in \mathbb{Z}^m$ *is a vector and* $t \in \mathbb{Q}$ *is a rational number.* $(\mathbf{B}, \mathbf{y}, t)$ *is a* YES *instance if* $\|\mathbf{Bx} - \mathbf{y}\| \leq t$ *for some* $\mathbf{x} \in \mathbb{Z}^n$. $(\mathbf{B}, \mathbf{y}, t)$ *is a* NO *instance if* $\|\mathbf{Bx} - \mathbf{y}\| > \gamma t$ *for all* $\mathbf{x} \in \mathbb{Z}^n$.

In our proof systems for lattice problems we make extensive use of a modular reduction technique proposed in [36] to emulate the effect of selecting a point uniformly at random from a lattice. Any lattice $\mathcal{L}(\mathbf{B})$ defines a natural equivalence relation on $\mathrm{span}(\mathbf{B}) = \sum_i \mathbf{b}_i \cdot \mathbb{R}$, where two points $\mathbf{x}, \mathbf{y} \in \mathrm{span}(\mathbf{B})$ are equivalent if $\mathbf{x} - \mathbf{y} \in \mathcal{L}(\mathbf{B})$. For any lattice basis $\mathbf{B}$ define the half open parallelepiped $\mathcal{P}(\mathbf{B}) = \{\mathbf{Bx} : 0 \leq x_i < 1\}$. It is easy to see that for any point $\mathbf{x} \in \mathrm{span}(\mathbf{B})$, there exists a unique point $\mathbf{y} \in \mathcal{P}(\mathbf{B})$ such that $\mathbf{x}$ is equivalent to $\mathbf{y}$ modulo the lattice. This unique representative for the equivalence class of $\mathbf{x}$ is denoted $\mathbf{x} \bmod \mathbf{B}$. Intuitively, $\mathbf{x} \bmod \mathbf{B}$ is the displacement of $\mathbf{x}$ within the fundamental parallelepiped containing $\mathbf{x}$. Notice that if we fix a (small) perturbation vector $\mathbf{r}$, we add it to a lattice point $\mathbf{Bv}$ and reduce the result modulo $\mathbf{B}$, we get a vector $(\mathbf{Bv} + \mathbf{r}) \bmod \mathbf{B} = \mathbf{r} \bmod \mathbf{B}$ that does not depend on the lattice point $\mathbf{Bv}$ from which we started. In other words, if we start from the origin, and simply compute $\mathbf{r} \bmod \mathbf{B}$, we obtain exactly the same distribution.

3 The Closest Vector Problem

In this section we describe a statistical zero-knowledge proof system (in fact, a proof of knowledge) with efficient provers for approximating the closest vector problem.

Consider an instance $(\mathbf{B}, \mathbf{y}, t)$ of GapCVP_γ. Look at a small ball around $\mathbf{y}$ and a small ball around a lattice point $\mathbf{Bw}$ closest to $\mathbf{y}$. If $\mathbf{y}$ and $\mathbf{Bw}$ are close to each other, the relative volume of the intersection of the two balls is quite large. So, if we pick a few random points from both balls, with high probability at least one of them will be in the intersection. The proof system works as follows: the prover picks random points from the two balls, reduces them modulo $\mathbf{B}$, and sends the reduced points to the verifier. Reducing the points modulo $\mathbf{B}$ has the nice effect that the resulting distribution can be efficiently sampled even without knowing the lattice point $\mathbf{Bw}$ closest to $\mathbf{y}$. (In fact, using two balls centered around $\mathbf{y}$ and the origin $\mathbf{0}$, results in exactly the same distribution after the reduction modulo $\mathbf{B}$. This is a crucial property to achieve zero-knowledge.) Let's say that the total number of points picked by the prover is even. Then, the verifier challenges the prover asking him to show that either (1) there is an even number of points from each ball; or (2) there is an odd number of points from each ball. If the prover can answer both challenges, then some point must belong to the intersection of the two balls, proving that the two balls intersect, and therefore their centers cannot be too far apart. Intuitively, the proof system is zero knowledge because all that the verifier sees is a set of random points from an efficiently samplable distribution.

Note that the proof system sketched above achieves neither perfect completeness nor perfect zero knowledge, but rather has a small (but negligible) completeness error and is *statistical* zero knowledge. The reason is that there is a nonzero probability that all the randomly chosen points will lie outside the intersection of the two balls, and in this case the prover will only be able to answer one of the two challenges. And intuitively, the verifier learns something in case the prover cannot answer, namely that none of the chosen points is in the intersection. Below, we achieve perfect completeness by having the prover modify the points chosen to ensure that at least one is in the intersection (if needed). However, this does not yield perfect zero knowledge, because now the points sent are no longer uniform in the two balls, but have a slightly skewed distribution that may be hard to sample exactly in polynomial time.

We now give the formal description of the proof system $(P_{\text{CVP}}, V_{\text{CVP}})$. In the description below k is a parameter to be determined that depends on the value of γ. In fact, the proof system is valid for any value of γ and k, and the choice of these parameters only affects the zero-knowledge property.

The Verifier. On input $(\mathbf{B}, \mathbf{y}, t)$, the verifier V_{CVP} proceeds as follows.

1. Receive k points $\mathbf{m}_1, \ldots, \mathbf{m}_k \in \mathbb{R}^n$ from the prover
2. Send a uniformly chosen random bit $q \in \{0, 1\}$ to the prover
3. Receive k bits $c_1, \ldots, c_k$ and k lattice points $\mathbf{Bv}_1, \ldots, \mathbf{Bv}_k$ and check that they satisfy $\sum_i c_i = q \pmod 2$ and $\|\mathbf{m}_i - (\mathbf{Bv}_i + c_i\mathbf{y})\| \le \gamma t/2$ for all i.

The following lemma shows that the protocol defined by the verifier is sound, both as an interactive proof system and even as a proof of knowledge.

Lemma 4 (soundness). *If $(\mathbf{B}, \mathbf{y}, t)$ is a* NO *instance of* GAPCVP_γ*, then the verifier V_{CVP} rejects with probability at least $1/2$ when interacting with any prover strategy P^*. Moreover, there is a probabilistic algorithm K (the* knowledge extractor*) such that if a prover P^* makes V_{CVP} accept with probability $1/2+\epsilon$ on some instance $(\mathbf{B}, \mathbf{y}, t)$, then $K^{P^*}(\mathbf{B}, \mathbf{y}, t)$ outputs a vector $\mathbf{w} \in \mathbb{Z}^n$ satisfying $\|\mathbf{Bw} - \mathbf{y}\| \leq \gamma t$ in expected time* $\text{poly}(n)/\epsilon$.

The Prover. Now that we know that the above proof system is sound, we show that if $(\mathbf{B}, \mathbf{y}, t)$ is a YES instance, then it is always possible to make the verifier accept. Suppose $(\mathbf{B}, \mathbf{y}, t)$ is a YES instance of GAPCVP_γ, i.e., there exists an integer vector $\mathbf{w} \in \mathbb{Z}^n$ such that $\|\mathbf{y} - \mathbf{Bw}\| \leq t$. We describe a probabilistic polynomial time prover P_{CVP} that, given the witness $\mathbf{w}$ (or, equivalently, $\mathbf{u} = \mathbf{y} - \mathbf{Bw}$) as auxiliary input, makes the verifier accept with probability 1. The prover P_{CVP}, on input $(\mathbf{B}, \mathbf{y}, t)$ and $\mathbf{u} = \mathbf{y} - \mathbf{Bw}$, proceeds as follows:

1. Choose $c_1, \ldots, c_k \in \{0, 1\}$ independently and uniformly at random. Also choose error vectors $\mathbf{r}_1, \ldots, \mathbf{r}_k \in \mathcal{B}(0, \gamma t/2)$ independently and uniformly at random. Then, check if there exists an index i^* such that $\|\mathbf{r}_{i^*} + (2c_{i^*} - 1)\mathbf{u}\| \leq \gamma t/2$. If not, set $i^* = 1$ and redefine $c_{i^*} = 0$ and $\mathbf{r}_{i^*} = \mathbf{u}/2$, so that $\|\mathbf{r}_{i^*} + (2c_{i^*} - 1)\mathbf{u}\| \leq \gamma t/2$ is certainly satisfied. Finally, compute points $\mathbf{m}_i = c_i\mathbf{y} + \mathbf{r}_i \bmod \mathbf{B}$ for all $i = 1, \ldots, k$ and send them to the verifier.
2. Wait for the verifier to reply with a challenge bit $q \in \{0, 1\}$.
3. If $q = \oplus_i c_i$, then the prover completes the proof sending bits c_i and lattice vectors $\mathbf{Bv}_i = \mathbf{m}_i - (\mathbf{r}_i + c_i\mathbf{y})$ (for $i = 1, \ldots, k$) to the verifier. If $q \neq \oplus_i c_i$, then the prover sends the same messages to the verifier, but with c_{i^*} and $\mathbf{Bv}_{i^*}$ replaced by $1 - c_{i^*}$ and $\mathbf{Bv}_{i^*} + (2c_{i^*} - 1)(\mathbf{y} - \mathbf{u})$.

It is clear that P_{CVP} can be implemented in polynomial time. The reader can easily verify that if the honest verifier V_{CVP} interacts with prover P_{CVP}, then it always accepts.

The Simulator. We prove the zero knowledge property by exhibiting a probabilistic polynomial-time simulator that outputs the transcript of a conversation between a (simulated) prover and a given cheating verifier V^* with a probability distribution that (for appropriate values of γ, k) is statistically close to that between V^* and the real prover P_{CVP}.

The simulator S_{CVP}, on input $(\mathbf{B}, \mathbf{y}, t)$, and given black-box access to a (possibly cheating) verifier V^*, proceeds as follows:

1. Pick random $c_1, \ldots, c_k \in \{0, 1\}$ and $\mathbf{r}_1, \ldots, \mathbf{r}_k \in \mathcal{B}(\mathbf{0}, \gamma t/2)$, and compute $\mathbf{m}_i = c_i\mathbf{y} + \mathbf{r}_j \bmod \mathbf{B}$ for all $i = 1, \ldots, k$.
2. Pass $\mathbf{m}_1, \ldots, \mathbf{m}_k$ to V^*, who replies with a query $q \in \{0, 1\}$.[9]

[9] We can assume, without loss of generality, that the verifier always output a single bit answer. Any other message can be interpreted in some standard way.

3. If $q = \oplus c_i$, then output the transcript $(\{\mathbf{m}_i\}_{i=1}^k, q, \{(c_i, \mathbf{B}\mathbf{v}_i)\}_{i=1}^k)$, where $\mathbf{B}\mathbf{v}_i = \mathbf{m}_i - (\mathbf{r}_i + c_i\mathbf{y})$. If $q \neq \oplus c_i$, then output `fail`.

Theorem 5. *If $(\mathbf{B}, \mathbf{y}, t)$ is a* YES *instance of* GAPCVP$_\gamma$*, then the statistical difference between the output of the simulator S_{CVP} (conditioned on the event that S_{CVP} does not fail), and the interaction between V^* and the real prover P_{CVP}, is at most $2(1 - \beta(2/\gamma))^k$, where $\beta(\epsilon)$ is the relative volume of the intersection of two unit spheres whose centers are at distance ϵ.*

Using the bound $\beta(\epsilon) \geq \max\left(\frac{3}{\exp(\epsilon^2 n/2)}, 1 - \epsilon\sqrt{n}\right)$ on the relative volume of the intersection of two spheres, [10] we immediately get the following corollary.

Corollary 6. *$(P_{\mathrm{CVP}}, V_{\mathrm{CVP}})$ is a statistical zero-knowledge proof system with perfect completeness and soundness error $1/2$, provided one of the following conditions holds true:*

- *$\gamma = \Omega(\sqrt{n/\log n})$ and $k = \mathrm{poly}(n)$ is a sufficiently large polynomial, or*
- *$\gamma = \Omega(\sqrt{n})$ and $k = \omega(\log n)$ is any superlogarithmic function of n, or*
- *$\gamma = n^{0.5+\Omega(1)}$ and $k = \omega(1)$ is any superconstant function of n.*

Negligible Error. As is, the proof system has constant soundness error (1/2), but it is often important to have negligible soundness error $(1/n^{\omega(1)})$. There are several approaches to reducing the soundness error, with different advantages:

(1) Repeat the proof system $\ell(n) = \omega(\log n)$ times in parallel. This unfortunately does not preserve the zero knowledge property, but does yield a constant-round statistically *witness-indistinguishable* proof of knowledge with negligible soundness error. (Witness indistinguishability means that for any two witness $\mathbf{w}$ and $\mathbf{w}'$, the verifier's view when the prover uses $\mathbf{w}$ is statistically close to its view when the prover uses $\mathbf{w}'$. See [34].)

(2) Repeat the proof system $\ell(n) = \Theta(\log n)$ times in parallel and then repeat the resulting protocol $\omega(1)$ times sequentially. This does preserve zero knowledge, yielding an $\omega(1)$-round statistical zero-knowledge proof of knowledge.

(3) In both of the approaches above, the ℓ-fold parallel repetition can be combined with the k-fold repetition already present in the original protocol to obtain more efficient protocols. Consider a modification of the original protocol $(P_{\mathrm{CVP}}, V_{\mathrm{CVP}})$, where in addition to sending k vectors in the first step, the prover also sends a random $k \times \ell$ matrix $\mathbf{M}$ over $\mathrm{GF}(2) = \{0, 1\}$. The verifier's challenge is then a random *vector* $\mathbf{q} \in \{0, 1\}^\ell$, and the condition $\oplus_i c_i = q$ is replaced with $\mathbf{Mc} = \mathbf{q}$. The advantage of this protocol is that it achieves both simulation and soundness error $2^{-\Omega(k)}$ with a protocol that involves only $O(k)$ n-dimensional vectors rather than $O(k^2)$ as achieved by independent repetitions of the original protocol.

[10] See [26] for a prove of the first inequality. The second one can be proved using similar techniques.

4 The Shortest Vector Problem

In this section we describe a statistical zero knowledge proof system $(P_{\mathrm{SVP}}, V_{\mathrm{SVP}})$ for GapSVP$_\gamma$. The reasons we are interested in the Shortest Vector Problem are both theoretical (being SVP a different problem from CVP, it is interesting to know if it admits **SZK** proofs with efficient prover), and practical, as proofs of knowledge for SVP can be used in conjunction with the lattices of [19] to yield identification schemes with worst-case/average-case security guarantees. (See Section 5.) Intuitively, our proof system for GapSVP can be thought as a combination of the reduction from GapSVP$_\gamma$ to GapCVP$_\gamma$ of Goldreich, Micciancio, Safra and Seifert [37], followed by the invocation of the proof system for GapCVP described in the previous section. Things are not as simple because the reduction of [37] is not a Karp reduction, and in order to solve a shortest vector problem instance, it requires the solution of (polynomially) many closest vector problems. So, we combine all the GapCVP instances together using the Goldreich-Levin hardcore predicate [38]. This is just the intuition behind the proof system that we are going to describe. In fact, our proof system requires neither the explicit construction of many GapCVP instances, nor the complicated analysis of the Goldreich-Levin predicate. So, below we briefly describe the proof system without reference to those general tools. For a detailed description see the full version of this paper.

The basic idea is the same as the proof system for the closest vector problem, but this time instead of selecting points close to the origin or close to the target vector $\mathbf{y}$, we consider balls centered around all lattice points of the form $\mathbf{Bc}$, where $\mathbf{c} \in \{0,1\}^n$, and reduce the points modulo $2\mathbf{B}$. The prover starts the interaction by sending points $\mathbf{m}_i$ close to randomly chosen centers $\mathbf{Bc}_i$. For each such point, the prover also sends a binary vector $\mathbf{s}_i$. If the lattice does not contain short vectors, then balls centered around different $\mathbf{Bc}$ are disjoint (even after reduction modulo $2\mathbf{B}$), and the first message sent by the prover uniquely determines a bit $\sum_i \langle \mathbf{s}_i, \mathbf{c}_i \rangle \bmod 2$. Then the verifier asks the prover to show that $\sum_i \langle \mathbf{s}_i, \mathbf{c}_i \rangle \bmod 2 = q$, where q is a random bit chosen by the verifier. If the prover can answer both questions, then there must be a message $\mathbf{m}_i$ that is close to two different centers (modulo $2\mathbf{B}$), proving that the lattice contains short vectors.

5 Identification Schemes

An *identification scheme* is a protocol by which one party, Alice, can repeatedly prove her identity to other parties in such a way that these parties cannot later impersonate Alice. Following the now-standard paradigm of [28], ID schemes are immediately obtained from zero-knowledge proofs of knowledge. It should be remarked that the computational problems underlying our identification schemes are not likely to be **NP**-hard (cf. [26]). The same is true for most computational problems used in cryptography (e.g., factoring), so, in some sense, ours is as good a hardness assumption as any. However, factoring is a much more widely studied assumption than lattice problems, so our identification schemes should be used with caution. The discussion below concentrates on efficiency issues.

The proofs of knowledge from Section 3, give rise to $\omega(1)$-round ID schemes, because witness-indistinguishability is not enough to guarantee the security. However, we can obtain a 3-round identification scheme as follows. First, we consider a new problem OR-GAPCVP$_\gamma$ whose instances are *pairs* (x_1, x_2) of GAPCVP$_\gamma$ instances, and whose YES instances are those for which at least one of the x_i's is a YES instance of GAPCVP$_\gamma$. Using a technique from [32], we can convert our proof system into one for OR-GAPCVP$_\gamma$. Parallel repetition yields a constant-round statistically witness-indistinguishable proof of knowledge with negligible soundness error. For such 'OR' problems, witness indistinguishability implies "witness hiding," which suffices for the identification scheme [39] (cf., [34]). Details will be given in the full version of the paper.

We stress that, unlike all known cryptosystems based on lattice problems [24, 23], these identification schemes only require the generation of lattices in which the approximate CLOSEST VECTOR PROBLEM (resp. SHORTEST VECTOR PROBLEM) is hard together with a close vector (resp. short vector). In particular, we do not need to generate an additional "short" basis, nor do we need "unique short vectors" or "hidden hyperplanes". In particular, this opens up more possibilities for using lattices with potential advantages both in terms of efficiency and security. As an example, for identification schemes based on SVP one can use the random class of lattices of [19, 22], which, for appropriate choice of the parameters, results in identification schemes that are at least as hard to break (on the average) as the worst case instance of approximating GAPSVP in the worst case within factor $\tilde{O}(n^4)$, or approximating other lattice problems (shortest linearly independent vectors or covering radius) within factor $\tilde{O}(n^3)$. Alternatively, one can use lattices with special structure like the cyclic and quasi-cyclic lattices of [20], or the convolutional modular lattices of [25] (but possibly with different, more secure, values of the parameters, since we do not need to embed a decryption trapdoor), in which the basis has a more compact representation (almost linear in the security parameter, rather than the standard matrix representation, whose quadratic size has been a practical barrier for the use of lattice cryptosystems.) Another very interesting possibility for identification schemes based on our CVP proof system is to use lattices where CVP *with preprocessing* (CVPP) is hard. This is a variant of the standard CVP problem introduced in [40] and studied in [41, 42], where finding close lattice vectors is hard even if the lattice is fixed, and the only input is the target vector. This allows to use the same lattice $\mathbf{B}$ for all users, and hardwire the description of the lattice $\mathbf{B}$ in the key generation, identification and verification algorithms. When a new user wants to generate a key, he chooses a random short error vector $\mathbf{r}$ (the secret key) and computes $\mathbf{y} = \mathbf{r} \bmod \mathbf{B}$ as its public key. The security of the scheme relies on the fact that approximating CVP in the lattice generated by $\mathbf{B}$ (for appropriately constructed, but fixed, $\mathbf{B}$) is hard. The advantage is that both the secret and public keys are just a single vector which takes storage proportional to dimension of the lattice n (the security parameter),[11] rather than a matrix

[11] This is obvious for the secret short vector $\mathbf{r}$. The public vector $\mathbf{y}$ can be much bigger because it contains large integer entries. Fortunately, as shown in [36], it is possible

(representing the lattice) which in general takes storage at least proportional to n^2. There are still big gaps between our understanding of CVPP and its cryptographic applicability: the strongest inapproximability results known to date [42] only show that CVPP is hard to approximate within factors smaller than 3, while our system requires inapproximability within $\sqrt{n}$. More importantly, all known lower bounds [40–42] only establish the hardness in the worst-case (**NP**-hardness), while for cryptographic applications one needs average-case hardness. Still, the possibility that further developments about the complexity of lattice problems might lead to *practical* and *provably secure* identification schemes with worst-case/average-case guarantees is very appealing. In this perspective, establishing a worst-case/average-case connection for CVPP along the lines of [19, 22] would be very interesting.

6 Statistical Difference

In this section, our focus will be the problem $\mathrm{SD}^{\alpha,\beta}$ for various values of $0 \leq \alpha < \beta \leq 1$. Consider the **SZK**-complete problem $\mathrm{SD}^{1/2,1-\epsilon}$, for $1/\sqrt{2} > \epsilon > 0$. Since we do not know if $\mathbf{SZK} \subseteq \mathbf{NP}$, we do not hope to give a proof system with efficient provers for this language. Instead we consider the limit problem $\mathrm{SD}^{1/2,1}$ obtained setting $\epsilon = 0$, i.e. deciding whether two distributions are statistically close or have disjoint supports. Unfortunately, this problem is not known to be complete for **SZK**. Note that $\mathrm{SD}^{1/2,1}$ is in **NP**, as coin tosses r_X, r_Y for which the circuits produce identical samples (i.e. $X(r_X) = Y(r_Y)$) are a witness that (X, Y) is a YES instance. We will prove that $\mathrm{SD}^{1/2,1}$ has a statistical zero-knowledge proof system with an efficient prover.

We now state a useful lemma that allows us to make the statistical difference exponentially small in YES instances of $\mathrm{SD}^{1/2,1}$.

Lemma 7 (XOR Lemma [8]). *Given probability distributions X_0, X_1 and a parameter k, define probability distributions $Y_c = (x_1, \ldots, x_k)$ (for $c \in \{0,1\}$) obtained by uniformly choosing $(b_1, \ldots, b_k) \leftarrow \{0,1\}^k$ such that $b_1 \oplus \cdots \oplus b_k = c$, and then sampling each $x_i \leftarrow X_{b_i}$ independently. Then $\Delta(Y_0, Y_1) = \Delta(X_0, X_1)^k$.*

Thus, given an instance (X_0, X_1) of $\mathrm{SD}^{1/2,1}$, this lemma shows how to construct circuits for a new pair of distributions (Y_0, Y_1) whose statistical difference is exponentially small if (X_0, X_1) is a YES instance, and whose supports are disjoint if (X_0, X_1) is a NO instance. We can use this to obtain simple statistical zero knowledge proof system for $\mathrm{SD}^{1/2,1}$, mimicking the well-known proof systems for QUADRATIC RESIDUOSITY [1] and GRAPH ISOMORPHISM [4]: **(1)** First, the prover sends the verifier $y \leftarrow Y_0$, **(2)** and the verifier replies sending $b \leftarrow \{0,1\}$ to the prover; **(3)** then the prover sends $r \leftarrow \{s : Y_b(s) = y\}$ to the verifier, **(4)** and the verifier accepts if $Y_b(r) = y$. It can be verified that the above proof system has soundness error $1/2$, completeness error $1/2^{k+1}$, and is

to select the basis $\mathbf{B}$ in an optimally secure way that results also in reduced vectors $\mathbf{y}$ with small bit-size.

statistical zero knowledge with simulator deviation $1/2^{k+1}$ (cf., [27]). However, even though $\mathrm{SD}^{1/2,1} \in \mathbf{NP}$, it does not appear that the prover strategy can be implemented in polynomial time given a witness. (If the verifier selects $b = 0$, the prover can respond with the coin tosses it used to generate y, but if the verifier selects $b = 1$, the prover must be able to find collisions between the circuits Y_0 and Y_1, which may be infeasible.)

To obtain efficient provers for $\mathrm{SD}^{1/2,1}$ itself, we use the ideas of Itoh, Ohta, and Shizuya [29]. The key concept is that of problem dependent commitment. This is a commitment scheme where the sender and receiver get as auxiliary input an instance x of a promise problem Π. The operations performed by the protocol depend on the value of x, and the protocol has different security properties depending on whether x is a YES or a NO instance of Π. Typically, the protocol is required to be secret when $x \in \Pi_{\mathrm{YES}}$ and unambiguous when $x \in \Pi_{\mathrm{NO}}$, or vice-versa. As usual, a problem dependent commitment is *statistically* secure if the secrecy and unambiguity properties hold in a statistical sense.

Itoh, Ohta, and Shizuya [29] considered only noninteractive problem-dependent commitment schemes in which both security properties are perfect. (Notice that any noninteractive statistically unambiguous commitment is necessarily perfectly unambiguous.) They proved that if a problem Π has a noninteractive problem-dependent commitment scheme which is perfectly secret on YES instances and perfectly unambiguous on NO instances, then Π has a perfect zero-knowledge proof system with an efficient prover. We observe that this result can be generalized as follows

Theorem 8 (generalizing [29]). *Suppose a promise problem Π is in* $\mathbf{NP}$*, with* $\mathbf{NP}$ *relation R, and that Π has a problem-dependent commitment scheme which is statistically secret on* YES *instances and statistically unambiguous on* NO *instances. Then Π has a statistical zero-knowledge proof system with an efficient prover (using any R-witness).*

We apply the theorem to $\mathrm{SD}^{1/2,1}$, by defining a problem dependent commitment for this problem as follows. On input $(b, (X_0, X_1), 1^k)$, the sender commits to b by sending the receiver $y \leftarrow Y_b$, where Y_b is obtained by applying the XOR Lemma (Lemma 7) to (X_0, X_1) with parameter k. In the reveal phase, the sender reveals b and the coin tosses used to generate y. The receiver checks that $Y_b(r) = y$. The reader can easily check that this commitment scheme is statistically secret on YES instances and perfectly unambiguous on NO instances. Using Theorem 8, we get the following result.

Theorem 9. $\mathrm{SD}^{1/2,1}$ *has a statistical zero-knowledge proof system with an efficient prover.*

6.1 Efficient provers for all of SZK?

As discussed in the introduction, part of our motivation in this work is the general question of whether every problem in $\mathbf{SZK} \cap \mathbf{NP}$ has a statistical zero-knowledge proof system with an efficient prover. The following theorem suggests three possible approaches to solve this problem.

Theorem 10. *If any of the following conditions hold, every problem in* **SZK** $\cap$ **NP** *has a statistical zero-knowledge proof with an efficient prover:*

1. $\mathrm{SD}^{1/3,2/3}$ *has a statistically secure problem-dependent commitment scheme.*
2. $\mathrm{SD}^{1/3,2/3}$ *reduces to* $\mathrm{SD}^{1/2,1}$ *via a randomized Karp reduction with* one-sided *error (even constant error).*
3. *Any* **NP** *problem that reduces to* $\mathrm{SD}^{1/3,2/3}$*, also reduces to* $\mathrm{SD}^{1/2,1}$.

The first approach is proved using the closure of **SZK** under complementation, and using the fact that if a promise problem Π reduces (via a randomized Karp reduction with one-sided *negligible* error probability) to a promise problem Γ, and Γ has a problem-dependent commitment scheme, then Π also has a problem-dependent commitment scheme with the same security properties. The second approach is just a way to prove the first condition, using the fact that one-sided error in Karp reductions to $\mathrm{SD}^{1/2,1}$ can be made negligible. The last approach, essentially asks to prove that $\mathrm{SD}^{1/2,1}$ is complete for **SZK** $\cap$ **NP**.

The proof systems described in this section differ in one important way from previous ones. All previous proof systems for variants of STATISTICAL DIFFERENCE e.g. [8, 9, 43], used the input circuits as "black boxes." That is, the use of the circuits by the verifier and prover consisted solely of evaluating the circuits on various inputs, and never referred to the internal structure of the circuits. It is not difficult to show, using constructions like [11], that no protocol of this form can be a statistical zero-knowledge proof with an efficient prover for even $\mathrm{SD}^{0,1}$ (if one-way functions exist). The proof system of Theorem 9 is not black box, however, and does make use of the internal workings of the circuits (due to the techniques of [29], which in turn use [4]). This suggests that this approach does indeed have potential to resolve questions that may have previously seemed intractable.

We conclude this section by showing yet another relationship between problem-dependent commitment schemes and **SZK**. If we remove the assumption that problem Π is in **NP** from the hypothesis of Theorem 8, we can still conclude that Π has an **SZK** proof system, although not necessarily one with efficient prover.

Proposition 11. *If a problem* Π *has a statistically secure problem-dependent commitment scheme, then* $\Pi \in$ **SZK**.

Acknowledgments

We are grateful to Oded Goldreich and Shafi Goldwasser for suggesting the problems on lattices. The second author thanks Danny Gutfreund, Moni Naor, and Avi Wigderson for discussions about **SZK** that influenced this work. We also thank Mihir Bellare for pointers to related work, and the anonymous reviewers and Minh Nguyen for helpful comments and corrections.

References

1. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM J. Comput. **18** (1989) 186–208
2. Bellare, M., Petrank, E.: Making zero-knowledge provers efficient. In: 24th STOC. (1992) 711–722
3. Boyar, J., Friedl, K., Lund, C.: Practical Zero-Knowledge Proofs: Giving Hints and Using Deficiencies. J. Cryptology **4** (1991) 185–206
4. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. J. ACM **38** (1991) 691–729
5. Fortnow, L.: The complexity of perfect zero-knowledge. In: Advances in Computing Research. Volume 5. JAC Press (1989) 327–343
6. Aiello, W., Håstad, J.: Statistical zero-knowledge languages can be recognized in two rounds. J. Comput. System Sci. **42** (1991) 327–345
7. Okamoto, T.: On relationships between statistical zero-knowledge proofs. J. Comput. System Sci. **60** (2000) 47–108
8. Sahai, A., Vadhan, S.: A complete problem for statistical zero knowledge. J. ACM **50** (2003) 196–249
9. Goldreich, O., Vadhan, S.: Comparing entropies in statistical zero-knowledge with applications to the structure of SZK. In: 14th CCC. (1999) 54–73
10. Goldreich, O., Sahai, A., Vadhan, S.: Honest verifier statistical zero-knowledge equals general statistical zero-knowledge. In: 30th STOC. (1998) 399–408
11. Vadhan, S.P.: On transformations of interactive proofs that preserve the prover's complexity. In: 32nd STOC. (2000) 200–207
12. Arvind, V., Köbler, J.: On pseudorandomness and resource-bounded measure. Theoret. Comput. Sci. **255** (2001) 205–221
13. Klivans, A.R., van Melkebeek, D.: Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. SIAM J. Comput. **31** (2002) 1501–1526
14. Miltersen, P.B., Vinodchandran, N.V.: Derandomizing Arthur-Merlin games using hitting sets. In: 40th FOCS. (1999) 71–80
15. Arora, S., Babai, L., Stern, J., Sweedyk, Z.: The hardness of approximate optima in lattices, codes, and systems of linear equations. J. Comput. System Sci. **54** (1997) 317–331
16. Ajtai, M.: The shortest vector problem in L_2 is NP-hard for randomized reductions (extended abstract). In: 30th STOC. (1998) 10–19
17. Micciancio, D.: The shortest vector problem is NP-hard to approximate to within some constant. SIAM J. Comput. **30** (2001) 2008–2035
18. Dinur, I., Kindler, G., Raz, R., Safra, S.: An improved lower bound for approximating CVP. Combinatorica (To appear) Preliminary version in FOCS '98.
19. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: 28th STOC. (1996) 99–108
20. Micciancio, D.: Generalized compact knapsaks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions (extended abstract). In: 43rd FOCS. (2002) 356–365
21. Goldreich, O., Goldwasser, S., Halevi, S.: Collision-free hashing from lattice problems. Technical Report TR96-056, ECCC (1996)
22. Micciancio, D.: Improved cryptographic hash functions with worst-case/average-case connection (extended abstract). In: 34th STOC. (2002) 609–618

23. Goldreich, O., Goldwasser, S., Halevi, S.: Public-key cryptosystems from lattice reduction problems. In: CRYPTO '97. Volume 1294 of Springer LNCS. (1997) 112–131
24. Ajtai, M., Dwork, C.: A public-key cryptosystem with worst-case/average-case equivalence. In: 29th STOC. (1997) 284–293
25. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: a ring based public key cryptosystem. In: Algorithmic number theory (ANTS III). Volume 1423 of Springer LNCS. (1998) 267–288
26. Goldreich, O., Goldwasser, S.: On the limits of nonapproximability of lattice problems. J. Comput. System Sci. **60** (2000) 540–563
27. Vadhan, S.P.: A Study of Statistical Zero-Knowledge Proofs. PhD thesis, MIT (1999)
28. Feige, U., Fiat, A., Shamir, A.: Zero-knowledge proofs of identity. J. Cryptology **1** (1988) 77–94
29. Itoh, T., Ohta, Y., Shizuya, H.: A language-dependent cryptographic primitive. J. Cryptology **10** (1997) 37–49
30. Gennaro, R., Micciancio, D., Rabin, T.: An efficient non-interactive statistical zero-knowledge proof system for quasi-safe prime products. In: 5th ACM CCS. (1998) 67–72
31. Tompa, M., Woll, H.: Random self-reducibility and zero knowledge interactive proofs of possession of information. In: 28th FOCS. (1987) 472–482
32. De Santis, A., Di Crescenzo, G., Persiano, G., Yung, M.: On monotone formula closure of SZK. In: 35th FOCS. (1994) 454–465
33. Bellare, M., Goldwasser, S.: The complexity of decision versus search. SIAM J. Comput. **23** (1994) 97–119
34. Goldreich, O.: Foundations of Cryptography: Basic Tools. Cambridge U. Press (2001)
35. Micciancio, D., Goldwasser, S.: Complexity of lattice problems: a cryptographic perspective. Volume 671 of Engineering and Computer Science. Kluwer (2002)
36. Micciancio, D.: Improving lattice based cryptosystems using the Hermite normal form. In: Cryptography and Lattices Conference. Volume 2146 of Springer LNCS. (2001) 126–145
37. Goldreich, O., Micciancio, D., Safra, S., Seifert, J.P.: Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. Inf. Proc. Lett. **71** (1999) 55–61
38. Goldreich, O., Levin, L.: A hard predicate for all one-way functions. In: 21st STOC. (1989)
39. Feige, U., Shamir, A.: Witness indistinguishable and witness hiding protocols. In: 22nd STOC. (1990) 416–426
40. Micciancio, D.: The hardness of the closest vector problem with preprocessing. IEEE Trans. Inform. Theory **47** (2001) 1212–1215
41. Feige, U., Micciancio, D.: The inapproximability of lattice and coding problems with preprocessing. J. Comput. System Sci. (To appear) Preliminary version in CCC 2002.
42. Regev, O.: Improved Inapproximability of Lattice and Coding Problems with Preprocessing. In: 18th CCC. (2003)
43. Goldreich, O., Sahai, A., Vadhan, S.: Can statistical zero-knowledge be made non-interactive?, or On the relationship of SZK and NISZK. In: CRYPTO '99. Volume 1666 of Springer LNCS. (1999)