Electronic Colloquium on Computational Complexity - Reports Series 1997 - available via:

Revision 02 of ECCC TR97-031

FTP: ftp.eccc.uni-trier.de:/pub/eccc/
WWW: http://www.eccc.uni-trier.de/eccc/
Email: ftpmail@ftp.eccc.uni-trier.de with subject 'help eccc'

On the Limits of Non-Approximability of Lattice Problems

Oded Goldreich*
Department of Computer Science
Weizmann Institute of Science
Rehovot, Israel.
oded@wisdom.weizmann.ac.il

Shafi Goldwasser†
Laboratory for Computer Science
Mass. Institute of Technology
Cambridge, MA02139.
shafi@theory.lcs.mit.edu

February 22, 1998

Abstract

We show simple constant-round interactive proof systems for problems capturing the approximability, to within a factor of $\sqrt{n}$, of optimization problems in integer lattices; specifically, the closest vector problem (CVP), and the shortest vector problem (SVP). These interactive proofs are for the "coNP direction"; that is, we give an interactive protocol showing that a vector is "far" from the lattice (for CVP), and an interactive protocol showing that the shortest-lattice-vector is "long" (for SVP). Furthermore, these interactive proof systems are Honest-Verifier Perfect Zero-Knowledge.

We conclude that approximating CVP (resp., SVP) within a factor of $\sqrt{n}$ is in $NP \cap \text{co}\mathcal{AM}$. Thus, it seems unlikely that approximating these problems to within a $\sqrt{n}$ factor is NP-hard. Previously, for the CVP (resp., SVP) problem, Lagarias *et. al.*, Håstad and Banaszczyk showed that the gap problem corresponding to approximating CVP (resp., SVP) within n is in $NP \cap \text{co}\mathcal{NP}$. On the other hand, Arora *et. al.* showed that the gap problem corresponding to approximating CVP within $2^{\log^{0.999} n}$ is quasi-NP-hard.

Keywords: Computational Problems in Integer Lattices, Hardness of Approximation, Interactive Proof Systems, AM, promise problems, smart reductions.

*Work done while visiting LCS, MIT.
†DARPA grant DABT63-96-C-0018.

1 Introduction

In recent years, many NP-hard optimization problems, have been shown to be hard to approximate as well. One current question of interest is how to know when the limit of inapproximability has been reached, and the problem becomes either tractable or at least not NP-hard to approximate. Two cases where the limits have been marked are the Min-Set-Cover problem and the Max-3SAT. For the Min-Set-Cover problem, the greedy approximation algorithm achieves a factor of approximation $\ln n$, whereas achieving any factor of approximation smaller than it is infeasible [16], unless $\mathcal{NP} \subseteq \widetilde{\mathcal{P}}$ (Quasi-Polynomial Time). For the Max-3SAT problem, a recent algorithm of [30] achieves an approximation ratio of $\frac{8}{7}$, whereas by [28] achieving any better factor of approximation would imply $\mathcal{NP} = \mathcal{P}$.

In this work, another possibility emerges as to how to show the limit of NP-Hardness of approximization. In particular, it is known that the Closest Vector Problem (CVP) is NP-Hard to approximate within any constant factor, and is infeasible to approximate within $2^{\log^{1-\epsilon} n}$ $(\forall \epsilon > 0)$ unless $\mathcal{NP}$ is in $\widetilde{\mathcal{P}}$ [6]. In this paper we show a constant-round interactive proof system for a (promise) problem capturing the approximation of CVP to within a factor of $\sqrt{n}$. This seems to indicate that it will be impossible to show an NP-Hardness type result for approximation factor $\sqrt{n}$. In particular, unless co$\mathcal{NP} \subseteq \mathcal{AM}$ (which in particular would collapse the Polynomial-Time Hierarchy [10]), such a result cannot be proven via a (randomized) many-to-one/Karp reduction. Furthermore, one would need to use a Turing/Cook reduction which makes queries outside of the promise – for further discussion see Section 5. We note that such reductions have not be used so far in the context of proving non-approximability results.

1.1 The computational problems considered

We consider two computational problems regarding integer lattices. The *closest vector problem* (CVP), and the *shortest vector problem* (SVP). In both cases, the dominant parameter is the dimension of the lattice, denoted n. The lattice is represented by a basis, denoted B, which is an n-by-n non-singular matrix over R. The lattice, $\mathcal{L}(B)$, is the set of points which can be expressed as integer linear combinations of the columns of B (i.e., $\mathcal{L}(B) \stackrel{\text{def}}{=} \{Bc : c \in \mathsf{Z}^n\}$).

The Closest Vector Problem (CVP). An input of the CVP problem consists of an n-dimensional lattice $\mathcal{L}$, and a target point t in R^n. The desired output is a point c in $\mathcal{L}$ which is closest to t (where 'closest' is defined with respect to a variety of norms l_p).

The CVP problem is NP-hard for all norms l_p, $p \geq 1$ (cf., van Emde Boas [40]). Furthermore, the problem is NP-hard to approximate within any constant factor (cf., [6]). The latter work also shows that if CVP could be approximated within any factor greater than $2^{\log^{1-\epsilon} n}$, then $\mathcal{NP} \subseteq \widetilde{\mathcal{P}}$. On the other hand Babai showed that CVP can be approximated within factor 2^n by a modification of the LLL lattice reduction algorithm.

The problem of *verifying* the "approximate-optimality" of a solution to the CVP problem has also been considered. Given a point c in the lattice, its distance to t clearly provides an upper bound on the minimum distance of t to the lattice, but there is no known way to verify in polynomial time that this distance in indeed minimal. Lagarias *et. al.* [33] showed, using reductions to the problem of computing Korkine–Zolotarev bases, that polynomial-size proofs exist that can be verified in polynomial-time that a vector c is within factor $n^{1.5}$ of the closest (to t) lattice vector. An improved bound of $O(n)$ was obtained by Håstad [27] and Banaszczyk [8], using dual lattices.

The Shortest Vector Problem (SVP). The SVP problem was formulated by Dirichlet in 1842. An input of the SVP problem is an n-dimensional lattice $\mathcal{L}$, and the desired output is a point c in $\mathcal{L}$ of minimum length (where 'length' is measured with respect to a variety of norms).[1]

The SVP problem has been known to be NP-hard in l_∞ (cf., [40]), and recently proved by Ajtai to be NP-hard (under randomized reductions) for the Euclidean l_2 norm [2]. Even more recently, Micciancio [36] has proven that it is NP-Hard (again under randomized reductions) to approximate the Shortest Vector Problem in l_2-norm to within any constant factor smaller than $\sqrt{2}$. The famous LLL lattice reduction algorithm [34] provides a polynomial-time approximation for SVP with an approximation factor of $2^{n/2}$, and improvements by [39] achieve for every $\epsilon > 0$ approximation within factor $2^{\epsilon n}$. No known results on hardness of approximation for SVP are known.

The problem of *verifying* the "approximate optimality" of a solution to the SVP problem has also been considered. The work of Lagarias *et. al.* [33] implies that polynomial-size proofs exist that can be verified in polynomial-time that a vector c in the lattice is within factor n of the shortest vector in the lattice. An alternative proof was suggested by Cai [12].

1.2 New Results: Short Interactive Proofs for approximate CVP and SVP

Hardness of approximation results for an optimization problem Φ are typically shown by reducing some hard problem (e.g., an NP-hard language) to a promise problem[2] related to the approximation of Φ. The approximation promise problem consists of a pair of subsets, $(\Pi_{\text{YES}}, \Pi_{\text{NO}})$, so that instances in Π_{YES} have a much "better value" than those in Π_{NO}. The gap between these values represents the approximation slackness, and distinguishing YES-instances from NO-instances captures the approximation task. In accordance with this methodology, which has been applied in all work regarding "hardness of approximation", we formulate promise problems capturing the approximation of CVP (resp., SVP) within a factor of $g(n)$.

Notation: By $\text{dist}(v, u)$ we denote the Euclidean distance between the vectors $v, u \in \mathsf{R}^n$. Extending this notation, we let $\text{dist}(V, U) \stackrel{\text{def}}{=} \min_{u \in U, v \in V}\{\text{dist}(v, u)\}$. In particular, we will be interested in $\text{dist}(v, \mathcal{L}(B))$, the distance of v from the lattice, $\mathcal{L}(B)$, spanned by the basis B.

The CVP promise problem (GapCVP): We consider the promise problem $\texttt{GapCVP}_g$, where g (the gap function) is a function of the dimension.

- YES instances (i.e., satisfying closeness) are triples (B, v, d) where B is a basis for a lattice in R^n, v is a vector in R^n, $d \in \mathsf{R}$ and $\text{dist}(v, \mathcal{L}(B)) \leq d$.
- NO instances (i.e., "strongly violating" closeness) are triples (B, v, d) where B is a basis for a lattice in R^n, $v \in \mathsf{R}^n$ is a vector, $d \in \mathsf{R}$ and $\text{dist}(v, \mathcal{L}(B)) > g(n) \cdot d$.

For any $g \geq 1$, the promise problem $\texttt{GapCVP}_g$ is in NP (i.e., in the extension of $\mathcal{NP}$ to promise problems): The NP-witness for (B, v, d) being a YES-instance is merely a vector $u \in \mathcal{L}(B)$ satisfying $\text{dist}(v, u) \leq d$. Also, by using the polynomial-time lattice reduction algorithms of [34, 39], we know that $\texttt{GapCVP}_{2^{\epsilon n}}$ is decidable in polynomial-time for every $\epsilon > 0$. No polynomial-time algorithm is known for smaller gap factors.

[1] An equivalent formulation used below refers to the minimum distance between a pair of distinct lattive points.

[2] A promise problem is a pair, $(\Pi_{\text{YES}}, \Pi_{\text{NO}})$, of non-intersecting subsets of $\{0,1\}^*$. The subset Π_{YES} (resp., Π_{NO}) corresponds to the YES-instances (resp., NO-instances) of the problem. The *promise* is the union of the two subsets; that is, $\Pi_{\text{YES}} \cup \Pi_{\text{NO}}$. Promise problems are a generalization of standard decision problems (i.e., language recognition problems) in which the promise holds for all strings (i.e., $\Pi_{\text{YES}} \cup \Pi_{\text{NO}} = \{0,1\}^*$).

Here we present a constant-round interactive proof system for the complement of the above promise problem with $g(n) = o(\sqrt{n})$. That is, we'll show that very-far instances (NO-instances) are always accepted, whereas close instances (YES-instances) are accepted with negligible probability. Specifically, we show that

Theorem 1.1 $\mathtt{GapCVP}_{\sqrt{n/O(\log n)}}$ *is in* co$\mathcal{AM}$.

Recall that by [33, 27, 8], $\mathtt{GapCVP}_n$ is in co$\mathcal{NP}$. Thus, we have placed a potentially harder problem (i.e., referring to smaller gaps) in a potentially bigger class (i.e., co$\mathcal{NP} \subseteq$ co$\mathcal{AM}$). Unlike the proofs of [33, 27, 8], which relies on deep results regarding lattices, our proof is totally elementary.

The SVP promise problem (GapSVP): We consider the promise problem $\mathtt{GapSVP}_g$, where g (the gap function) is again a function of the dimension.

- YES instances (i.e., having short vectors) are pairs (B, d) where B is a basis for a lattice $\mathcal{L}(B)$ in R^n, $d \in \mathsf{R}$ and $\mathrm{dist}(v_1, v_2) \leq d$ for some $v_1 \neq v_2$ in $\mathcal{L}(B)$.
- NO instances (i.e., "strongly violating" short vectors) are pairs (B, d) where B and d are as above but $\mathrm{dist}(v_1, v_2) > g(n) \cdot d$ for all $v_1 \neq v_2$ in $\mathcal{L}(B)$.

Again, for any $g \geq 1$, the promise problem $\mathtt{GapSVP}_g$ is in NP, the problem $\mathtt{GapCVP}_{2^{\epsilon n}}$ is decidable in polynomial-time (for every $\epsilon > 0$), but no polynomial-time algorithm is known for smaller gap factors.

We present a constant-round interactive proof system for the complement of the above promise problem with $g(n) = o(\sqrt{n})$. That is, we'll show that NO-instances are always accepted, whereas YES-instances are accepted with negligible probability.

Theorem 1.2 $\mathtt{GapSVP}_{\sqrt{n/O(\log n)}}$ *is in* co$\mathcal{AM}$.

Recall that by [33], $\mathtt{GapCVP}_n$ is in co$\mathcal{NP}$. Again, in contrast to [33], our proof is elementary.

On the complexity of unique-SVP: Using our results, Cai has recently proved that the following promise problem, called $f(n)$-unique SVP, is in co$\mathcal{NP} \cap \mathcal{AM}$ for $f(n) = \sqrt[4]{n/O(\log n)}$. The input to the problem is a pair (B, v), and the promise is that the shortest vector in $\mathcal{L}(B)$, denoted u, is $f(n)$-unique in the sense that for every $u' \in \mathcal{L}(B)$ if $\|u'\| \leq f(n) \cdot \|u\|$ then u' is an integer multiple of u. The problem is to distinguish the case when v is the shortest vector of $\mathcal{L}(b)$ from the case it is not. Cai (cf., [12]) has shown a many-to-one reduction of $f(n)$-unique SVP to the complement of $\mathtt{GapSVP}_g$, for $g(n) = f(n) \cdot \sqrt{f(n)^2 - 0.25}$ (which is approximately $f(n)^2$, provided $f(n) = \omega(1)$).

Comment on Zero-Knowledge: Our constant-round interactive proofs for the complement of $\mathtt{GapCVP}_{\sqrt{n/O(\log n)}}$ and the complement of $\mathtt{GapSVP}_{\sqrt{n/O(\log n)}}$ are actually Perfect Zero-Knowledge (PZK) with respect to an Honest Verifier. Using recent results regarding zero-knowledge proof systems [37, 38, 21], it follows that both these problems as well as their complements have (general) Statistical Zero-Knowledge proof systems (i.e., are in SZK).[3]

[3] Specifically, Honest-Verifier Statistical Zero-Knowledge (SZK) proofs (of which Honest-Verifier PZK is a special case) are closed under complementation [37], and this holds also for promise problems [38]. Furthermore, Honest-Verifier SZK proofs can be transformed into ones of the public-coin type [37], and by a recent result of [21] the latter can be transformed into general SZK proofs (i.e., robust against any verifier strategy).

Comment on other norms: Our proof systems can be adapted to any l_p norm (and in particular to l_1 and l_∞). Specifically, we obtain constant-round (HVPZK) interactive proof systems for gap $n/O(\log n)$ (rather than gap $\sqrt{n/O(\log n)}$ as in l_2 norm). The result extend to any *computationally tractable* norm as defined in Section 4. (Except for Section 4, the rest of the paper refers to CVP and SVP in l_2 norm.)

Comment on computational problems regarding Linear Codes: Our proof systems can be easily adapted to the corresponding Nearest and Lightest codeword problems for linear codes.[4] In both cases the obtained gap is $n/O(\log n)$, where n is the length of the codewords. As suggested by Madhu Sudan (priv. comm. 1997), for the Nearest codeword problem, a similar bound can be obtained by using the standard reduction of the coding problem to CVP in l_1 norm.

1.3 Implication on proving non-approximability of CVP and SVP

In [20], the existence of an AM-proof system for Graph Non-Isomorphism (GNI) was taken as evidence to the belief that Graph Isomorphism (GI) is unlikely to be $\mathcal{NP}$-complete. The reasoning was that a reduction (even a Cook reduction) of $\mathcal{NP}$ to GI would imply that co$\mathcal{NP}$ is in $\mathcal{AM}$, and thus that the Polynomial-Time Hierarchy collapses [10].

We have to be more careful when promise problems are concerned. If $\mathcal{NP}$ is KARP-reducible to $\text{GapCVP}_{\sqrt{n}}$ (or to any promise problem in $\mathcal{NP} \cap \text{co}\mathcal{AM}$) then it follows that co$\mathcal{NP} \subseteq \mathcal{AM}$. However it is not clear what happens (in general) if $\mathcal{NP}$ is COOK-reducible to a promise problem in $\mathcal{NP} \cap \text{co}\mathcal{AM}$. The difficulty is with the case in which the Cook reduction makes some queries for which the promise does not hold. For such a query the validity of the answer is not necessarily provable via an AM system. Thus, $\mathcal{NP}$ may be COOK-reducible to a promise problem in $\mathcal{NP} \cap \text{co}\mathcal{AM}$ and still co$\mathcal{NP} \subseteq \mathcal{AM}$ may not hold. In fact, Even *et. al.* [15, Thm. 4] constructed an NP-Hard promise problem in $\mathcal{NP} \cap \text{co}\mathcal{NP}$ (and co$\mathcal{NP} \subseteq \mathcal{NP}$ does not seem to follow). Restricting our attention to *smart reductions* [25], Cook reductions for which all queries satisfy the promise, we show that if $\mathcal{NP}$ is reducible to a promise problem in $\mathcal{NP} \cap \text{co}\mathcal{AM}$ via a smart reduction, then co$\mathcal{NP} \subseteq \mathcal{AM}$.

Our results thus imply that (at least) ONE of the following three MUST HOLD:

1. (*Most Probable*): $\text{GapCVP}_{\sqrt{n}}$ is NOT $\mathcal{NP}$-hard.

2. $\text{GapCVP}_{\sqrt{n}}$ is $\mathcal{NP}$-hard but with a reduction which is NOT many-to-one and furthermore makes queries which violate the promise.

3. (*Most improbable*): co$\mathcal{NP} \subseteq \mathcal{AM}$ and in particular the Polynomial-Time Hierarchy collapses.

Ruling out the third possibility, we view our results as establishing limits on results regarding the hardness of approximating CVP and SVP: Approximations to within a factor of $\sqrt{n}$ are either not NP-hard or their NP-hardness must be established via reductions which make queries violating the promise (of the target promise problem). See Section 5 for further discussion.

We note that Arora *et. al.* [6] have essentially conjectured that $\text{GapCVP}_{\sqrt{n}}$ is $\mathcal{NP}$-hard. The above can be taken as evidence that the conjecture is false.

Remark: We note that in discussions in the litreature (cf. [6]), the result of Lagarias *et. al.* [33] is taken mistakenly to mean that approximating CVP within $n^{1.5}$ cannot be NP-hard, unless co$\mathcal{NP} \subseteq \mathcal{NP}$. The possibility of NP-Hardness via non-smart Cook-reductions is ignored, although it does

[4]This fact, not stated in our preliminary posting on ECCC, was discovered independently by Alekhnovich [4].

apply there as well. What can be said is that [33] implies that a proof that approximaing CVP within $n^{1.5}$ is NP-Hard either will employ non-smart Cook-reductions or would imply that co$\mathcal{NP} \subseteq \mathcal{NP}$.

The cryptographic angle: Interest in the difficulty of $\mathtt{GapCVP}$ and $\mathtt{GapSVP}$ has increased recently as versions of both has been suggested as basis for Cryptographic primitives and schemes (cf., [1, 19, 3]). In particular, in a pioneering work [1],[5] Ajtai has constructed a one-way function assuming that $\mathtt{GapSVP}_{n^c}$ is hard (in worst case), where $c > 11$.[6] Ajtai and Dwork [3] proposed a public-key encryption scheme whose security is reduced to a special case of (a search version of) $\mathtt{GapSVP}_{n^c}$ (with some big c). Interestingly, the trapdoor permutation suggested in [19] relies on the conjectured difficulty of the Closest Vector Problem. On the other hand, $\mathtt{GapCVP}_{2^{\log^{1-\epsilon} n}}$ is quasi-NP-hard [6], and $\mathtt{GapSVP}_{\sqrt{2}-\epsilon}$ is NP-hard [2, 36], for any $\epsilon > 0$. An immediate question which arises is whether the security of a cryptographic system can be based on the difficulty of $\mathtt{GapCVP}_{g(n)}$ or $\mathtt{GapSVP}_{g(n)}$ for a function g for which these approximation problems are NP-hard (or, say, quasi-NP-hard). Our results indicate that $g(n)$ may need be $o(\sqrt{n}/\log n)$.

The above raises again an old question, regarding the possibility – in general – of basing the security of cryptosystems on the assumption that $\mathcal{P} \neq \mathcal{NP}$. We discuss this question in Section 6.

2 (HVPZK) constant-round proof for "non-closeness"

We consider the promise problem $\mathtt{GapCVP}_g$ defined in the introduction, and present a constant-round interactive proof system for the complement of the above problem for gap $g(n) = \sqrt{n/O(\log n)}$. Recall that the input is a triple (B, v, d), where B is a basis for a lattice, v is a vector and $d \in \mathsf{R}$. That is, we'll show that NO-instances (in which v is at distance greater than $g(n) \cdot d$ from the lattice) are always accepted, whereas YES-instances (in which v is within distance d from $\mathcal{L}(B)$) are accepted with probability bounded away from 1.

The proof system: Consider a "huge" sphere, denoted H. Specifically, we consider a sphere of radius $2^n \cdot \|(B, v)\|$ centered at the origin, where $\|(B, v)\|$ denotes the length of the largest vector in $B \cup v$. Let $g = g(n)$.

1. The verifier uniformly selects $\sigma \in \{0, 1\}$, a random lattice point in H, denoted r, and an error vector, η, uniformly distributed in a sphere of radius $gd/2$. The verifier sends $x \stackrel{\text{def}}{=} r + \sigma v + \eta$ to the prover.

2. The prover responses with $\tau = 0$ if $\mathrm{dist}(x, \mathcal{L}(B)) < \mathrm{dist}(x, \mathcal{L}(B) + v)$ and $\tau = 1$ otherwise, where $U + v \stackrel{\text{def}}{=} \{u + v : u \in U\}$.

3. The verifier accepts if and only if $\tau = \sigma$.

Implementation details. Several obvious implementation questions, arising from the above description, are

- *How to uniformly select a lattice point in H?* We uniformly select a point in H, represent this point as a linear combination of the basis vectors, and obtain a lattice point by rounding. This procedure partitions H into cells, most of them are parallelepipeds which are isomorphic

[5] The fundamental aspect of that work, not discussed here, is the reduction of a worst-case problem to an average-case one.

[6] The constant has been recently reduced to $c > 5$ by Cai and Nerurkar [13].

to the basic cell/parallelepiped defined by the lattice. The exceptions are the partial parallelepipeds which are divided by the boundary of the sphere H. All the latter parallelepipeds are contained between two co-centered spheres, the larger being of radius $(2^n + n) \cdot L$ and the smaller being of radius $(2^n - n) \cdot L$, where $L \stackrel{\text{def}}{=} \|(B, v)\| \geq \|B\|$ is the radius of H. Thus, the fraction of these ("divided") parallelepipeds in the total number of parallelepipeds is bounded above by the volume encompassed between the above two spheres divided by the volume of the smaller sphere. This relative volume is at most

$$\begin{aligned}\frac{(2^n + n)^n - (2^n - n)^n}{(2^n - n)^n} &= \left(1 + \frac{2n}{2^n - n}\right)^n - 1 \\ &< \frac{3n^2}{2^n}\end{aligned}$$

It follows, that the above procedure generates random lattice points in a distribution which is at most $\mathrm{poly}(n) \cdot 2^{-n}$ away from the uniform distribution over $\mathcal{L}(B) \cap H$.

- *How to uniformly select a point in the unit sphere?* One may just invoke the general algorithm of Dyer *et. al.* [14]. Using this algorithm, it is possible to select almost uniformly a point in any convex body (given by a membership oracle). Alternatively, one may select the point by generating n samples from the standard normal distribution, and normalize the result so that a vector of length r appears with probability proportional to r^{-n} (see, e.g., [32, Sec. 3.4.1]).
- *How to deal with finite precision?* In the above description, we assume all operations to be done with infinite precision. This is neither possible nor needed. We assume, instead, that the input entries (in the vectors), are given in rational representation and let m denote the number of bits in the largest of the corresponding integers. Then making all calculations with $n^3 \cdot m$ bits of precision, introduces an additional stochastic deviation of less than 2^{-n} in our bounds.

Analysis of the protocol. By the above, it should be clear that the verifier's actions in the protocol can be implemented in probabilistic polynomial-time. We will show that, for $g(n) = \sqrt{n/O(\log n)}$, the above protocol constitutes a (Honest Verifier Perfect Zero-Knowledge) proof system for the promise problem $\texttt{GapCVP}_g$, with perfect completeness and soundness error bounded away from 1.

Claim 2.1 (perfect completeness): *If* $\mathrm{dist}(v, \mathcal{L}(B) > g(n) \cdot d$ *then the verifier always accepts* (when interacting with the prover specified above).

Proof: Under the above hypothesis, for every point x (and in particular the messages sent by verifier in Step 1), we have $\mathrm{dist}(x, \mathcal{L}(B)) + \mathrm{dist}(x, \mathcal{L}(B) + v) > gd$ (or else $\mathrm{dist}(v, \mathcal{L}(B)) = \mathrm{dist}(\mathcal{L}(B) + v, \mathcal{L}(B)) \leq \mathrm{dist}(x, \mathcal{L}(B) + v) + \mathrm{dist}(x, \mathcal{L}(B)) \leq dg$). Thus, for every message, $x = r + \sigma v + \eta$, sent by the verifier we have

$$\begin{aligned}\mathrm{dist}(x, \mathcal{L}(B) + \sigma v) &= \mathrm{dist}(r + \eta, \mathcal{L}(B)) \leq \|\eta\| \leq \frac{dg}{2} \\ \mathrm{dist}(x, \mathcal{L}(B) + (1 - \sigma) \cdot v) &> gd - \mathrm{dist}(x, \mathcal{L}(B) + \sigma v) \geq \frac{dg}{2}\end{aligned}$$

Thus, it is always the case that $\mathrm{dist}(x, \mathcal{L}(B) + \sigma v) < \mathrm{dist}(x, \mathcal{L}(B) + (1 - \sigma) \cdot v)$ and the prover responses with $\tau = \sigma$. ■

Claim 2.2 (zero-knowledge): *The above protocol is perfect (honest-verifier) zero-knowledge over triples (v, B, d) satisfying* $\mathrm{dist}(v, \mathcal{L}(B)) > g(n) \cdot d$.

Proof: The simulator just reads the verifier's choice and returns it as the prover's message. Thus, the simulator's output will consist of coins for the verifier and the prover's response. By the above proof, this distribution is identical the verifier's view in the real protocol. ■

Claim 2.3 (soundness): *Let $c > 0$ and $g(n) \geq \sqrt{\frac{n}{c \ln n}}$, if* $\mathrm{dist}(v, \mathcal{L}(B)) \leq d$ *then, no matter what the prover does, the verifier accepts with probability at most* $1 - n^{-2c}$.

The above is slightly inaccurate as the statement holds only for sufficiently large n's (depending on the constant c). For smaller (fixed) dimension, one may replace the protocol by an immediate computation using Lenstra's algorithm [35]. The same holds for Claim 3.3 below.

2.1 Proof of the soundness claim

Let ξ_0 (resp., ξ_1) a random variable representing the message sent by the verifier condition on $\sigma = 0$ (resp., $\sigma = 1$). Below, we upper bound the statistical distance between the two random variables by $(1 - 2n^{-2c})$. Given this bound, we have for any prover strategy P^*

$$\begin{aligned}
\Pr(P^*(\xi_\sigma) = \sigma) &= \frac{1}{2} \cdot \Pr(P^*(\xi_0) = 0) + \frac{1}{2} \cdot \Pr(P^*(\xi_1) = 1) \\
&= \frac{1}{2} + \frac{1}{2} \cdot (\Pr(P^*(\xi_0) = 0) - \Pr(P^*(\xi_1) = 0)) \\
&\leq \frac{1}{2} + \frac{1}{2} \cdot (1 - 2n^{-2c}) \\
&= 1 - n^{-2c}
\end{aligned}$$

Thus, all that remains is to prove the above bound on the statistical distance between ξ_0 and ξ_1. The statistical distance between the two random variables is due to two sources:

1. In case $\sigma = 1$ the point $r + v$ may be out of the sphere H (whereas, by choice, r is alway in H). However, since H is much bigger than v this happens rarely (i.e., with probability at most $3n^2 \cdot 2^{-n}$; see above). Furthermore, the statistical difference between uniform distribution on the lattice points in the sphere H and the same distribution shifted by adding the vector v is negligible. Specifically, we may bound it by $n^{-2c} > 3n^2 \cdot 2^{-n}$.

2. Let v' represent the shortest vector leading from the lattice to the point v (i.e., $v - v' \in \mathcal{L}(B)$ so that $\|v'\| \leq d$). For each lattice point, p, we consider the statistical distance between $p + \eta$ and $p + v' + \eta$, where η is as above. This is the main source of statistical distance between ξ_0 and ξ_1, and the rest of the proof is devoted to upper bound it.

It suffices to consider the statistical distance between η and $v' + \eta$, where η is as above. In the first case the probability mass is uniformly distributed in a sphere of radius $gd/2$ centered at 0^n, whereas in the second case the probability mass is uniformly distributed in a sphere of radius $gd/2$ centered at v'. Without loss of generality, we consider $v' = (d, 0, ..., 0)$. Normalizing things (by division with $gd/2$), it suffices to consider the statistical distance between the following two distributions:

(D1) Uniform distribution in a unit sphere centered at the origin.

(D2) Uniform distribution in a unit sphere centered at point $(\epsilon, 0, .., 0)$, where $\epsilon = \frac{d}{gd/2} = \frac{2}{g}$.

Observe that the statistical distance between the two distributions equals *half* the volume of the symmetric difference of the two spheres divided by the volume of a sphere. Thus, we are interested in the relative symmetric difference of the two spheres. Recall two basic facts –

Fact 2.4 (e.g., [5, Vol. 2, Sec. 11.33, Ex. 4]): *The volume of an n-dimensional sphere of radius r is* $v_n(r) \stackrel{\text{def}}{=} \frac{\pi^{n/2}}{\Gamma((n/2)+1)} \cdot r^n$, *where* $\Gamma(x) = (x-1) \cdot \Gamma(x-1)$, $\Gamma(1) = 1$, $\Gamma(0.5) = \sqrt{\pi}$.

Fact 2.5 (e.g., [31, Sec. 1.2.11.2, Exer. 6]): *For sufficiently large real* $x > 2$, $\Gamma(x+1) \approx \sqrt{2\pi x} \cdot (x/e)^x$. *Thus, for sufficiently large integer,* $m > 2$,

$$\frac{\Gamma(m+0.5)}{\Gamma(m)} \approx \sqrt{m} \approx \frac{\Gamma(m+1)}{\Gamma(m+0.5)}$$

Lemma 2.6 (symmetric difference between close spheres): *Let S_0 (resp. S_ϵ) be a unit sphere at the origin (at distance ϵ from the origin). Then relative symmetric difference between the spheres (i.e., the symmetric difference divided by the volume) is at most*

$$2 - \epsilon \cdot \frac{(1-\epsilon^2)^{(n-1)/2}}{3} \cdot \sqrt{n}$$

Our bound is not tight. Still, we note that the bound cannot be decreased below $2-(1-(\epsilon/2)^2)^{(n-1)/2} \cdot \sqrt{n}$, and that both expressions are equivalent as far as our application goes.

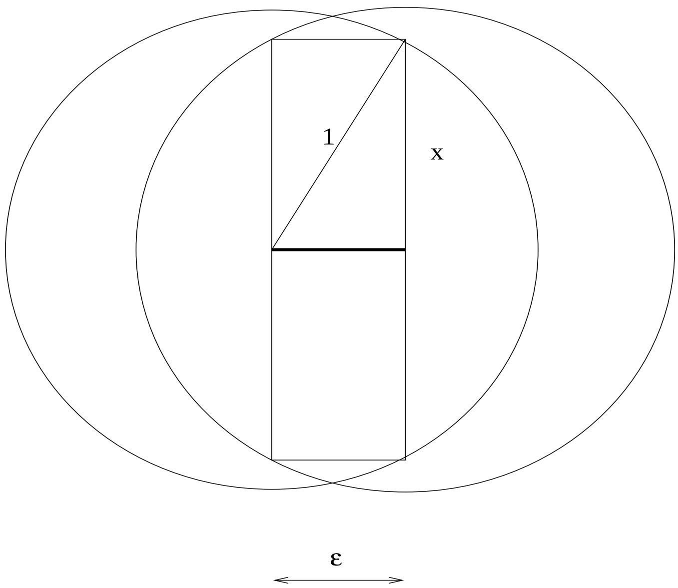

Figure 1: The cylinder encompassed by S_0 and S_ϵ. The axis is marked in bold and its radius $x = (1-\epsilon^2)^{0.5}$ is computed from the center of the left sphere.

Proof: We will lower the volume of the intersection between S_0 and S_ϵ. Specifically, we look at the $(n-1)$-dimensional cylinder of height ϵ, which is centered at the axis connecting the centers of

S_0 and S_ϵ and in encompassed by $S_0 \cap S_\epsilon$. See Figure 2.1. The radius of this cylinder is $\sqrt{1-\epsilon^2}$. Thus its volume is $\epsilon \cdot v_{n-1}(\sqrt{1-\epsilon^2})$. Using Facts 2.4 and 2.5 we have

$$\begin{aligned}
\frac{\mathrm{vol}(S_0 \cap S_\epsilon)}{\mathrm{vol}(S_0)} &> \frac{\epsilon \cdot v_{n-1}(\sqrt{1-\epsilon^2})}{v_n(1)} \\
&= \frac{\epsilon \cdot (1-\epsilon^2)^{(n-1)/2} \cdot v_{n-1}(1)}{v_n(1)} \\
&= \epsilon \cdot (1-\epsilon^2)^{(n-1)/2} \cdot \frac{\Gamma((n/2)+1)}{\sqrt{\pi} \cdot \Gamma((n/2)+0.5)} \\
&\approx \epsilon \cdot (1-\epsilon^2)^{(n-1)/2} \cdot \frac{\sqrt{n/2}}{\sqrt{\pi}}
\end{aligned}$$

The lemma follows. ■

Using Lemma 2.6, with $\epsilon = \frac{2}{g(n)} \leq \sqrt{\frac{4c \ln n}{n}}$, we upper bound the statistical distance between distributions (D1) and (D2) by

$$\begin{aligned}
\frac{1}{2} \cdot \left(2 - \epsilon\sqrt{n} \cdot \frac{(1-\epsilon^2)^{(n-1)/2}}{3}\right) &\leq 1 - \frac{\sqrt{4c \ln n}}{6} \cdot \left(1 - \frac{4c \ln n}{n}\right)^{(n-1)/2} \\
&< 1 - \frac{\sqrt{c \ln n}}{3} \cdot \left(1 - \frac{2c \ln n}{n/2}\right)^{n/2} \\
&< 1 - 3 \cdot n^{-2c}
\end{aligned}$$

where the last inequality uses $\sqrt{c \ln n} > 9$. Thus, the statistical distance between ξ_0 and ξ_1 is bounded by $n^{-2c} + 1 - 3 \cdot n^{-2c}$ (where the extra n^{-2c} term comes from Item 1 above). The soundness claim follows. ■

2.2 Conclusion

Combining the above protocol with known transformations (i.e., [24] and [7]), we get

Theorem 1 *The promise problem* $\mathtt{GapCVP}_{\sqrt{n/O(\log n)}}$ *is in* $\mathcal{NP} \cap \mathrm{co}\mathcal{AM}$. *Furthermore, the complement of* $\mathtt{GapCVP}_{\sqrt{n/O(\log n)}}$ *has a HVPZK constant-round proof system.*

The interesting part is the membership of $\mathtt{GapCVP}_{\sqrt{n}}$ in co$\mathcal{AM}$. This reduces the gap factor for which "efficient proof systems" exists: Lagarias *et. al.* [33], Håstad [27] and Banaszczyk [8] have previously shown that $\mathtt{GapCVP}_n$ is in co$\mathcal{NP}$.

3 (HVPZK) constant-round proof for "no short-vector"

We consider the promise problem $\mathtt{GapSVP}_g$ defined in the introduction, and present a constant-round interactive proof system for the complement of the above problem for gap $g(n) = \sqrt{n/O(\log n)}$. Recall that the input is a pair (B, d), where B is a basis for a lattice and $d \in \mathsf{R}$. That is, we'll show that NO-instances (in which the shortest vector in $\mathcal{L}(B)$ has length greater than $g(n) \cdot d$) are always accepted, whereas YES-instances (in which $\mathcal{L}(B)$ has a non-zero vector of length at most d) are accepted with probability bounded away from 1.

The proof system: Consider a huge sphere, denoted H (as in Section 2). Specifically, we consider a sphere of radius $2^n \cdot \|B\|$ centered at the origin. Let $g = g(n)$.

1. The verifier uniformly selects a random lattice point, p, in H, and an error vector, η, uniformly distributed in a sphere of radius $gd/2$. The verifier sends $\widetilde{p} \stackrel{\text{def}}{=} p + \eta$ to the prover.

2. The prover sends back the closest lattice point to $\widetilde{p}$.

3. The verifier accepts iff the prover has answered with p.

Claim 3.1 (perfect completeness): *If every two distinct lattice points are at distance greater than gd then the verifier always accepts.*

Proof: Under the above hypothesis, for every point x (and in particular the message sent by verifier in step 1), we have at most one lattice vector v so that $\text{dist}(x, v) \leq gd/2$ (or else $\text{dist}(v_1, v_2) \leq \text{dist}(x, v_1) + \text{dist}(x, v_2) \leq gd$). Since we have $\text{dist}(\widetilde{p}, p) \leq gd/2$, the prover always returns p, where p and $\widetilde{p}$ are as in Step 1. ∎

Claim 3.2 (zero-knowledge): *The above protocol is perfect (honest-verifier) zero-knowledge over pairs (B, d) for which every two distinct points in $\mathcal{L}(B)$ are at distance greater than gd.*

Proof: The simulator just reads the verifier's choice and returns it as the prover's message. Thus, the simulator's output will consist of coins for the verifier and the prover's response. By the above proof, this distribution is identical the verifier's view in the real protocol. ∎

Claim 3.3 (soundness): *Let $c > 0$ and $g(n) \geq \sqrt{\frac{n}{c \ln n}}$, if for some $v_1 \neq v_2$ in $\mathcal{L}(B)$, $\text{dist}(v_1, v_2) \leq d$ then, no matter what the prover does, the verifier accepts with probability at most $1 - n^{-2c}$.*

Proof: Let $p' \stackrel{\text{def}}{=} p + (v_1 - v_2)$, where p is the lattice point chosen by the verifier in Step 1. Clearly, $\text{dist}(p, p') \leq d$. Let ξ be a random variable representing the message actually sent by the verifier, and let $\xi' = \xi + (v_1 - v_2)$. Using the analysis in the proof of Claim 2.3, we bound the statistical distance between these two random variables by $(1 - 3n^{-2n})$. (Note that ξ corresponds to ξ_0 and ξ' corresponds to ξ_1 with $v' = v_1 - v_2$.) Given this bound, we have for any prover strategy P^*

$$\begin{aligned} \Pr(P^*(\xi) = p) &\leq (1 - 3n^{-2n}) + \Pr(P^*(\xi') = p) \\ &\leq 2 - 3n^{-2n} - \Pr(P^*(\xi') = p') \end{aligned}$$

However, the event $P^*(\xi') = p'$ is almost as probable as $P^*(\xi) = p$ (with the only difference in probability due to the case where p' is outside the sphere which happens with probability at most n^{-2n}). Thus, we have

$$\begin{aligned} 2 \cdot \Pr(P^*(\xi) = p) &< \Pr(P^*(\xi) = p) + \Pr(P^*(\xi') = p') + n^{-2n} \\ &\leq 2 - 2n^{-2n} \end{aligned}$$

and the claim follows. ∎

Conclusion: Combining the above protocol with known transformations (i.e., [24] and [7]), we get

Theorem 2 *The promise problem* $\texttt{GapSVP}_{\sqrt{n/O(\log n)}}$ *is in* $\mathcal{NP} \cap \mathrm{co}\mathcal{AM}$. *Furthermore, the complement of* $\texttt{GapSVP}_{\sqrt{n/O(\log n)}}$ *has a HVPZK constant-round proof system.*

Again, the interesting part is the membership of $\texttt{GapSVP}_{\sqrt{n}}$ in co$\mathcal{AM}$. This reduces the gap factor for which "efficient proof systems" exists: Lagarias *et. al.* [33] have previously shown that $\texttt{GapSVP}_n$ is in co$\mathcal{NP}$.

4 Treating other norms

The underlying ideas of Theorems 1 and 2 can be applied to provide (HVPZK) constant-round proof systems for corresponding gap problems regarding any "computationally tractable" norm and in particular for all ℓ_p-norms (e.g., the ℓ_1 and ℓ_∞ norms). The gap factor is however larger: $n/O(\log n)$ rather than $\sqrt{n/O(\log n)}$.

Tractable norms: Recall the norm axioms (for a generic norm $\|\cdot\|$) –

(N1) For every $v \in \mathsf{R}^n$, $\|v\| \geq 0$, with equality holding if and only if v is the zero vector.

(N2) For every $v \in \mathsf{R}^n$ and any $\alpha \in \mathsf{R}$, $\|\alpha v\| = |\alpha| \cdot \|v\|$.

(N3) For every $v, u \in \mathsf{R}^n$, $\|v + u\| = \|v\| + \|u\|$. (Triangle Inequality).

To allow the verifier to conduct is actions in polynomial-time, we make the additional two requirements

(N4) The norm function is polynomial-time computable. That is, there exist a polynomial-time algorithm that, given a vector v and an accuracy parameter δ, outputs a number in the interval $[\|v\| \pm \delta]$. We stress that the algorithm is uniform over all dimensions.

(N5) The unit sphere defined by the norm contains a ball of radius $2^{-\mathrm{poly}(n)}$ centered at the origin, and is contained in a ball of radius $2^{\mathrm{poly}(n)}$ centered at the origin. That is, there exists a polynomial p so that for all n's,

$$\{v \in \mathsf{R}^n : \|v\|_2 \leq 2^{-p(n)}\} \subseteq \{v \in \mathsf{R}^n : \|v\| \leq 1\} \subseteq \{v \in \mathsf{R}^n : \|v\|_2 \leq 2^{p(n)}\}$$

where $\|v\|_2$ is the Euclidean (ℓ_2) norm of v.

Note that axioms (N4) and (N5) are satisfied by all (the standard) ℓ_p-norms.[7] On the other hand, by [14], axioms (N4) and (N5) suffice for constructing a probabilistic algorithm which given n, generates in time poly(n) a vector which is almost uniformly distributed in the n-dimensional unit sphere w.r.t the norm. Specifically, by axioms (N2) and (N3), the unit sphere is a convex body, and axioms (N4) and (N5) imply the existence of a so-called "well-guaranteed weak membership oracle" (cf., [26]) as required by the convex body algorithm of Dyer *et. al.* [14] (and its improvements – e.g., [29]).

[7] Actually, for any ℓ_p-norm, there is a simple algorithm for uniformly selecting a point, $(x_1, ..., x_n)$, in the corresponding unit sphere: Generate n independent samples, $x_1, ..., x_n$, each with density function e^{-x^p}, and normalize the result so that a vector of norm r appears with probability proportional to r^{-n}.

Our protocols can be adapted to any norm satisfying the additional axioms (N4) and (N5). We modify the protocols of the previous sections so that the error vector, η, is chosen uniformly among the vectors of norm less than $g(n)d/2$ (rather than being chosen uniformly in a sphere of radius $g(n)d/2$). Here we use $g(n) \stackrel{\text{def}}{=} n/O(\log n)$. Clearly the completeness and zero-knowledge claims continue to hold as they merely relied on the triangle inequality (i.e., Norm axiom (N3)). In the proof of the soundness claim, we replace Lemma 2.6 by the following lemma in which distance refers to the above norm (rather than to Euclidean norm):

Lemma 4.1 (symmetric difference between close spheres, general norm): *For every $c > 0$, let p be a point at distance $\epsilon < 1$ from the origin. Then the relative symmetric difference between the set of points of distance 1 from the origin and the set of points of distance 1 from p is at most $2 \cdot (1 - (1 - \epsilon)^n)$.*

We comment that the bound is quite tight for both the ℓ_1 and the ℓ_∞ norm. That is, in both cases the relative symmetric difference is at least $2 - (1 - (\epsilon/2))^n$.[8]

Proof: Let B_0^r (resp., B_p^r) denote the set of points of distance r from the origin (resp., from p). The symmetric difference between B_0^1 and B_p^1 equals twice the volume of $B_p^1 \setminus B_0^1$. This volume is clearly bounded above by $B_p^1 \setminus B_p^{1-\epsilon}$. By the norm axioms (N1) and (N2), we have $\frac{\mathrm{vol}(B_p^1 \setminus B_p^{1-\epsilon})}{\mathrm{vol}(B_p^1)} = 1 - (1-\epsilon)^n$, and the lemma follows. ■

Using $\epsilon = \frac{2}{g(n)}$ and $g(n) = n/O(\log n)$, we conclude that the proof system has soundness error bounded above by $1 - (1 - \frac{O(\log n)}{n})^n = 1 - \frac{1}{\mathrm{poly}(n)}$. Repeating it polynomially many times in parallel we get

Theorem 3 *Both* `GapCVP` *and* `GapSVP` *defined for any norm and gap factor $n/O(\log n)$ are in $\mathcal{NP} \cap \mathrm{co}\mathcal{AM}$. Furthermore, the complement promise problems have HVPZK constant-round proof systems.*

5 What does it mean?

To simplify the discussion we extend the definition of standard complexity classes to promise problem. For example, a promise problem $\Pi = (\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}})$ is said to be in $\mathcal{NP}$ if there exists a polynomial-time recognizable (witness) relation R so that

- For every $x \in \Pi_{\mathrm{YES}}$ there exists a $y \in \{0,1\}^*$ such that $(x, y) \in R$ (and $|y| = \mathrm{poly}(|x|)$).
- For every $x \in \Pi_{\mathrm{NO}}$ and every $y \in \{0,1\}^*$, $(x, y) \notin R$.

As stated in the Introduction, the fact that a promise problem in $\mathcal{NP} \cap \mathrm{co}\mathcal{NP}$ (resp., $\mathcal{AM} \cap \mathrm{co}\mathcal{AM}$) is NP-hard *via arbitrary Cook reductions* does not seem to imply that $\mathcal{NP} = \mathrm{co}\mathcal{NP}$ (resp., $\mathrm{co}\mathcal{NP} \subseteq \mathcal{AM}$). However, such a conclusion does hold in case NP-hardness is proven by a restricted type of Cook-reductions, called *smart reductions* and defined by Grollmann and Selman.

[8] To verify the above claim for ℓ_∞, consider the point $p = (\epsilon, \epsilon, ..., \epsilon)$. Clearly, the intersection of the unit sphere centered at the origin and the unit sphere centered at p is $(2 - \epsilon)^n$, whereas each sphere has volume 2^n. For ℓ_1, consider the point $p = (\epsilon, 0, ..., 0)$. Again, the intersection is a sphere of radius $1 - (\epsilon/2)$ (according to the norm in consideration).

Definition 4 (smart reduction [25]): *A* **smart reduction** *of a promise problem A to a promise problem B is a polynomial-time* (possibly randomized) *Cook-reduction that on input which satisfies the promise of A only makes queries which satisfy the promise of B. Otherwise the reduction is called* **non-smart**.[9]

We note that any many-to-one/Karp (possibly randomized) reduction is smart, and that all known inapproximability results were proven via such reductions of $\mathcal{NP}$ to a corresponding gap problem (such as `GapCVP`). On the other hand, Grollmann and Selman proved [25, Thm. 2] that if a $\mathcal{NP}$-complete language has an smart reduction to a promise problem in $\mathcal{NP} \cap \mathrm{co}\mathcal{NP}$ then $\mathcal{NP} = \mathrm{co}\mathcal{NP}$. It is quite straightforward to adapt their argument to obtain –

Theorem 5 *Suppose that a $\mathcal{NP}$-complete language has an smart reduction to a promise problem in $\mathcal{AM} \cap \mathrm{co}\mathcal{AM}$. Then $\mathrm{co}\mathcal{NP} \subseteq \mathcal{AM}$.*

Proof: Given any co$\mathcal{NP}$-language L, we use the smart (deterministic) reduction to the promise problem Π in order to construct an AM-proof system for L. On input x, the prover sends to the verifier a transcript of an accepting computation of the reduction (i.e., the oracle-machine). This transcript includes queries to the Π-oracle and presumed answers of this oracle. Next, the prover proves that each of these answers is correct by running the adequate AM-proof system (for either Π or its complement). Here we use the hypothesis that the reduction is smart (which implies that the prover can always succeed in case $x \in L$). We stress that all these AM-proofs are run in parallel, and so the result is an MAM-proof system (which can be converted into an AM-proof system [7]). In case of a randomized (smart) reduction, we let the verifier select the random input (to the reduction) and continue as above. $\blacksquare$

Corollary 6 *If either* $\mathtt{GapCVP}_{\sqrt{n}}$ *or* $\mathtt{GapSVP}_{\sqrt{n}}$ *is $\mathcal{NP}$-hard* via smart reductions *then* $\mathrm{co}\mathcal{NP} \subseteq \mathcal{AM}$.

It is known that the CVP is NP-Hard to approximate within any constant factor, and is hard to approximate within $2^{\log^{1-\epsilon} n}$ unless $\mathcal{NP}$ is in $\widetilde{\mathcal{P}}$ (Quasi-Polynomial time) [6]. (Both reductions are many-to-one.) Arora *et. al.* [6] set as a challenge to prove that $\mathtt{GapCVP}_{\sqrt{n}}$ is $\mathcal{NP}$-hard. The corollary above, however, can be taken as evidence of the impossibility of proving NP-Hardness result for approximation factor below $\sqrt{n}$ for CVP or SVP. Specifically, unless $\mathrm{co}\mathcal{NP} \subseteq \mathcal{AM}$, such a result will have to be derived via a non-smart Cook reduction. We note that such reductions have not be used so far towards proving in-approximability results.

6 On the possibility of basing Cryptography on the assumption that $\mathcal{P} \neq \mathcal{NP}$

The discussion of the "cryptographic angle" in the introduction raises again an old question:

> Is it possible to base the security of cryptosystems on the difficulty of NP-hard problems.

A claim of *impossiblity* is commonly attributed to Brassard. However, what Brassard actually showed [11, Thm. 2, Item (2)ii] can be stated as follows

[9] Unfortunately, the term "non-smart" is somewhat misleading – to be non-smart (in an essential way) and yet work the reduction must be quite "clever". A term like "safe" or "honest" may have been more suitable than smart; however "honest" is taken and using "safe" may be confusing when talking about cryptography.

Brassard's Claim: *Consider a public-key encryption scheme with a* deterministic *encryption algorithm, and suppose that the set of valid public-keys is in* co$\mathcal{NP}$. *Then, if the problem of retrieving the plaintext from the* (ciphertext, public-key) *pair is NP-Hard, then it follows that* $\mathcal{NP} = \mathrm{co}\mathcal{NP}$.

There are two problem with the hypothesis of this *impossibility* result, aside from the well known fact that worst-case hardness of retrieving the plaintext is an inadequate notion of security of encryption schemes. The problems are, firstly, that the encryption algorithm is postulated to be deterministic, and secondly that the set of valid public-keys for it is postulated to form a co$\mathcal{NP}$-set. While these preconditions are satisfied in certain encryption schemes (and in particular in the schemes known at the time the claim was made, e.g., plain RSA), they are *not* satisfied in probabilistic encryption schemes such as the Goldwasser–Micali [22] and the Blum–Goldwasser scheme [9] (as well as to the recent "lattice-based" schemes of [3, 19]). We mention that probabilistic encryption is essential to security as defined in [22].

Thus, Brassard's Claim does not rule out the possibility of "basing cryptography" (or even public-key encryption) on the assumption that $\mathcal{P} \neq \mathcal{NP}$ (even if $\mathcal{NP} \neq \mathrm{co}\mathcal{NP}$, as we do believe). Furthermore, such a possibility is not ruled out even by extentions of Brassard's Claim of which we are aware (cf., [18]), and which do cover some probabilistic encryption schemes (such as the abovementioned [22, 9]).

Acknowledgments

We are grateful to Mihir Bellare, Jin-Yi Cai, Yevgeniy Dodis, Shai Halevi, Johan Håstad, Ravi Kannan, László Lovász, Moni Naor, Muli Safra, Jean-Pierre Seifert, Alan Selman, Adi Shamir and Madhu Sudan for helpful discussions.

References

[1] M. Ajtai. Generating Hard Instances of Lattice Problems. In *28th STOC*, pages 99–108, 1996.

[2] M. Ajtai. The Shortest Vector Problem in L_2 is NP-Hard for Randomized Reductions. In *30th STOC*, 1998.

[3] M. Ajtai and C. Dwork. A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence, In *29th STOC*, pages 284–293, 1997.

[4] M. Alekhnovich. On approximating the Minimal Code Distance. Private communication, Oct. 1997.

[5] T.M. Apostol. *Calculus*, Vol. 2 (second edition). John Wiley & Sons, Inc., 1969.

[6] S. Arora, L. Babai, J. Stern and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *Journal of Computer and System Sciences*, Vol. 54, pages 317–331, 1997.

[7] L. Babai. Trading Group Theory for Randomness. In *17th STOC*, pages 421–420, 1985.

[8] W. Banaszczyk. New Bounds in some Transference Theorems in the Geometry of Numbers. *Mathematiche Annalen*, 296, pages 625–635 (1993).

[9] M. Blum and S. Goldwasser. An Efficient Probabilistic Public-Key Encryption Scheme which hides all partial information. In *Crypto84*, LNCS (196) Springer-Verlag, pages 289–302.

[10] R. Boppana, J. Håstad, and S. Zachos. Does Co-NP Have Short Interactive Proofs? *IPL*, 25, May 1987, pp. 127-132.

[11] G. Brassard. Relativized Cryptography. In *20th FOCS*, pages 383–391, 1979.

[12] J. Cai. A relation of primal-dual lattices and the complexity of shortest lattice vector problem. To appear in *TCS*.

[13] J. Cai and A.P. Nerurkar. An improved Worst-Case to Average-Case connection for lattice problems. In *38th FOCS*, pages 468–477, 1997.

[14] M. Dyer, A. Frieze and R. Kannan. A Random Polynomial-Time Algorithm for Approximating the Volume of Convex Bodies. *Journal of the ACM*, Vol. 38, pages 1–17, 1991.

[15] S. Even, A.L. Selman, and Y. Yacobi. The Complexity of Promise Problems with Applications to Public-Key Cryptography. *Inform. and Control*, Vol. 61, pp. 159–173, 1984.

[16] U. Feige. A threshold of ln n for approximating set cover. In *28th STOC*, pages 314–318, 1996.

[17] L. Fortnow. The Complexity of Perfect Zero-Knowledge. In *Advances in Computing Research: a research annual*, Vol. 5 (Randomness and Computation, S. Micali, ed.), pages 327–343, 1989.

[18] O. Goldreich and S. Goldwasser. On the possibility of basing Cryptography on the assumption that $\mathcal{P} \neq \mathcal{NP}$. Manuscript, Feb. 1998.

[19] O. Goldreich, S. Goldwasser and S. Halevi. Public-Key Cryptosystems from Lattice Reduction Problems. In *Crypto97*, Springer LNCS, Vol. 1294, pp. 112–131.

[20] O. Goldreich, S. Micali and A. Wigderson. Proofs that Yield Nothing But Their Validity or All Languages in NP Have Zero-Knowledge Proof Systems. *JACM*, Vol. 38, No. 1, pp. 691–729, 1991.

[21] O. Goldreich, A. Sahai, and S. Vadhan. Honest-Verifier Statistical Zero-Knowledge equals general Statistical Zero-Knowledge. In *30th STOC*, 1998.

[22] S. Goldwasser and S. Micali. Probabilistic Encryption. *JCSS*, Vol. 28, No. 2, pages 270–299, 1984.

[23] S. Goldwasser, S. Micali and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM J. Comput.*, Vol. 18, No. 1, pp. 186–208, 1989.

[24] S. Goldwasser and M. Sipser. Private Coins versus Public Coins in Interactive Proof Systems. In *18th STOC*, pages 59–68, 1986.

[25] J. Grollmann and A.L. Selman. Complexity Measures for Public-Key Cryptosystems. *SIAM J. Comput.*, Vol. 17, No. 2, pages 309–335, 1988.

[26] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer–Verlag, 1988.

[27] J. Håstad. Dual Vectors and Lower Bounds for the Nearest Lattice Point Problem. *Combinatorica*, Vol. 8, 1988, pages 75–81.

[28] J. Håstad. Getting optimal in-approximability results. In *29th STOC*, pages 1–10 1997.

[29] R. Kannan, L. Lovász and M. Simonovits. Random walks and $O^*(n^5)$ volume algorithm for convex bodies. Preprint, 1997. To appear in *Random Structures and Algorithms.*

[30] H. Karloff and U. Zwick. A 7/8-eps approximation algorithm for MAX 3SAT? In *38th FOCS*, pages 406–415, 1997.

[31] D.E. Knuth. *The Art of Computer Programming*, Vol. 1 (second edition). Addison–Wesley Publishing Company, Inc., 1973.

[32] D.E. Knuth. *The Art of Computer Programming*, Vol. 2 (second edition). Addison–Wesley Publishing Company, Inc., 1981.

[33] J. Lagarias, H.W. Lenstra, C.P. Schnorr. Korkine–Zolotarev bases and successive minima of a lattice and its reciprocal lattice. *Combinatorica*, Vol. 10, pages 333-348, 1990.

[34] A.K. Lenstra, H.W. Lenstra, L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen* 261, 515-534 (1982).

[35] H.W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, Vol. 8, pages 538–548, 1983.

[36] D. Micciancio. On the Inapproximability of the Shortest Vector in a Lattice within some constant factor. Preliminary version MIT/LCS/TM-574, February 1998.

[37] T. Okamoto. On relationships between statistical zero-knowledge proofs. In *28th STOC*, pages 649–658, 1996.

[38] A. Sahai and S. Vadhan. A Complete Promise Problem for Statistical Zero-Knowledge. In *38th FOCS*, pages 448–457, 1997.

[39] C.P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. In *Theoretical Computer Science*, vol. 53, 1987, pp. 201-224

[40] P. van Emde Boas. Another NP-complete problem and the complexity of computing short vectors in a lattice. Report 81-04, Mathematische Instituut, Uni. Amsterdam, 1981.