On Basing Search SIVP on NP-Hardness

Tianren Liu(✉)

MIT, Cambridge, USA
liutr@mit.edu

Abstract. The possibility of basing cryptography on the minimal assumption $\mathbf{NP} \not\subseteq \mathbf{BPP}$ is at the very heart of complexity-theoretic cryptography. The closest we have gotten so far is lattice-based cryptography whose average-case security is based on the worst-case hardness of approximate shortest vector problems on integer lattices. The state-of-the-art is the construction of a one-way function (and collision-resistant hash function) based on the hardness of the $\tilde{O}(n)$-approximate shortest independent vector problem $\mathsf{SIVP}_{\tilde{O}(n)}$.

Although SIVP is **NP**-hard in its exact version, Guruswami et al. (CCC 2004) showed that $\mathsf{gapSIVP}_{\sqrt{n/\log n}}$ is in $\mathbf{NP} \cap \mathbf{coAM}$ and thus unlikely to be **NP**-hard. Indeed, any language that can be reduced to $\mathsf{gapSIVP}_{\tilde{O}(\sqrt{n})}$ (under general probabilistic polynomial-time adaptive reductions) is in $\mathbf{AM} \cap \mathbf{coAM}$ by the results of Peikert and Vaikuntanathan (CRYPTO 2008) and Mahmoody and Xiao (CCC 2010). However, none of these results apply to reductions to *search problems*, still leaving open a ray of hope: *can* **NP** *be reduced to solving search SIVP with approximation factor* $\tilde{O}(n)$?

We eliminate such possibility, by showing that any language that can be reduced to solving search SIVP with any approximation factor $\lambda(n) = \omega(n \log n)$ lies in **AM** intersect **coAM**.

1 Introduction

It is a long-standing open question whether cryptography can be based on the minimal assumption that $\mathbf{NP} \not\subseteq \mathbf{BPP}$. More precisely, one would hope to construct cryptographic primitives such that given a polynomial-time algorithm breaking the security of the primitive, one can efficiently solve SAT.

The closest we have gotten so far is lattice cryptography. This approach was born out of the breakthrough result of Ajtai [Ajt96], which constructs a one-way function family based on the *worst-case* hardness of certain lattice problems such as the γ-approximate shortest independent vectors problem (SIVP_γ), which can be stated as follows: given an n-dimensional lattice, find a set of n linearly independent vectors whose length[1] is at most $\gamma(n)$ (polynomial in n) times the length of the shortest such vector set. Since the work of Ajtai, the state of the

Research supported in part by NSF Grants CNS-1350619 and CNS-1414119.

[1] The length of a vector set is defined as the length of the longest vector in the set.

A. Beimel and S. Dziembowski (Eds.): TCC 2018, LNCS 11239, pp. 98–119, 2018.
https://doi.org/10.1007/978-3-030-03807-6_4

art is a construction of a family of collision resistant hash functions (CRHF) based on the hardness of the shortest independent vectors problem with an approximation factor $\tilde{O}(n)$ [MR04]. One would hope that this approach is viable for constructing cryptography based on **NP**-hardness since Blömer and Seifert showed that SIVP_γ is **NP**-hard for any constant factor [BS99]. Presumably, if one could construct cryptographic primitives based on the hardness of $\mathsf{SIVP}_{O(1)}$, we would be golden. Alternatively, if one could extend the result of Blömer and Seifert to show the **NP**-hardness of SIVP_γ for larger $\gamma(n)$, we would be closer to the goal of basing cryptography on **NP**-hardness.

However, there are some negative results when one considers the corresponding gap version of the same lattice problem. The gap problem, denoted by $\mathsf{gapSIVP}_\gamma$, is to estimate the length of the short independent vector set within a factor of $\gamma(n)$. Peikert and Vaikuntanathan show that $\mathsf{gapSIVP}_{\omega(\sqrt{n \log n})}$ is in **SZK** [PV08]. Thus there is no Cook reduction from SAT to $\mathsf{gapSIVP}_{\tilde{O}(\sqrt{n})}$ unless the polynomial hierarchy collapses (as $\mathbf{BPP}^{\mathbf{SZK}} \subseteq \mathbf{AM} \cap \mathbf{coAM}$ [MX10]).

Fortunately, the hardness of SIVP is not contradicted by the fact that the gap problem with the same approximation factor is easy. For instance, if one considers any ideal lattice in the field $\mathbb{Z}[x]/\langle x^{2^k} + 1\rangle$, its successive minima satisfy $\lambda_1 = \ldots = \lambda_n$, thus $\mathsf{gapSIVP}_{\sqrt{n}}$ can be trivially solved using Minkowski's inequality. However, finding a set of short independent vectors in such ideal lattices is still believed to be hard. As none of these negative results apply to reductions to search SIVP, there is still a ray of hope: *can* **NP** *be reduced to solving search* SIVP *with approximation* $\tilde{O}(n)$?

Thus, in order to really understand the viability of the approach begun by the work of Ajtai, it seems one must study the search versions of lattice problems. In this work, we relate the hardness of the search version SIVP_γ, to the gap version gapSIVP. Informally, we show that if $\mathsf{gapSIVP}_\gamma$ is not hard, neither is $\mathsf{SIVP}_{\sqrt{n}\cdot\gamma}$.

Main Theorem 1. *If* $\mathsf{gapSIVP}_\gamma \in \mathbf{SZK}$ *and there exists a probabilistic polynomial-time adaptive reduction from a language* L *to* $\mathsf{SIVP}_{\sqrt{n \log n}\cdot\gamma}$*, then* $\mathsf{L} \in \mathbf{AM} \cap \mathbf{coAM}$.

As a quick corollary, combining our result with $\mathsf{gapSIVP}_{\omega(\sqrt{n \log n})} \in \mathbf{SZK}$ [PV08], any language that can be reduced to $\mathsf{SIVP}_{\omega(n \log n)}$ lies in **AM** intersect **coAM** and thus it is not **NP**-hard unless the polynomial hierarchy collapses.

Corollary 1.1. *If there exists a probabilistic polynomial-time adaptive reduction from a language* L *to* SIVP_γ *for any* $\gamma(n) = \omega(n \log n)$*, then* $\mathsf{L} \in \mathbf{AM} \cap \mathbf{coAM}$.

1.1 Proof Overview

The first step is to shift from a search problem to a sampling problem. Our goal is to obtain a black-box separation between SIVP_γ and **NP**-hardness by showing that any language L that can be reduced to SIVP_γ is in **AM** intersect **coAM**. Let $\mathcal{R}$ be the reduction from L to SIVP_γ. We will construct an **AM** protocol for L using reduction $\mathcal{R}$. For a first attempt, the naïve verifier samples a random tape

and sends it to the prover. The prover simulates the reduction $\mathcal{R}$ and resolves any query to SIVP_γ using its unbounded computational power. The simulation, including the answers to the reduction's query to SIVP_γ, is sent to the naïve verifier, so that the verifier can check its correctness. But SIVP_γ is a search problem and there is no unique right answer. The prover has the freedom to decide which answer is chosen upon each query. This freedom allows a malicious prover to fool the naïve verifier. Similar difficulty were faced by Bogdanov and Brzuska, which is resolved by inherently shifting to sampling problems. In order to separate size-verifiable one-way functions from **NP**-Hardness [BB15], they force the prover to sample a random answer uniformly among all correct ones. Thus the correct answer distribution for each query is unique.

Inspired by the work of Bogdanov and Brzuska, we consider a sampling problem related to SIVP_γ, called the discrete Gaussian distribution. A discrete Gaussian over a lattice is a distribution such that the probability of any vertex $\mathbf{v}$ is proportional to $e^{-\pi\|\mathbf{v}-\mathbf{c}\|^2/s^2}$, where $\mathbf{c}$ is its "center" and parameter s is its "width". Lemma 4.3 shows that discrete Gaussian sampling is as hard as SIVP_γ in the sense that there is a black-box reduction from SIVP_γ to discrete Gaussian sampling with "width" $\gamma(n)/\sqrt{n}$. Therefore, if language L can be reduced to SIVP_γ, then it can also be reduced to discrete Gaussian sampling on lattices with "width" $s \leq \lambda_n/\sqrt{n}$.

Lemma 4.3 (Informal). SIVP_γ *can be efficiently reduced to discrete Gaussian sampling on lattices with "width"* $\sigma = \frac{\gamma}{\sqrt{n}}\lambda_n$.

Lemma 4.3 is a generalization of [Reg09, Lemma 3.17]. Its proof is quite intuitive. Repeatedly sample from the discrete Gaussian over the same lattice centered at $\mathbf{0}$. With good probability, the newly sampled vertex is short and is linearly independent from previously sampled verteces.

The next natural question is, *which property separates a sampling problem from* **NP***-hardness?* Here we introduce the notion of "probability-verifiability". Informally, a distribution family is *probability-verifiable* if for any distribution $\mathcal{D}$ in this family and for any possible value v, $\Pr[v \leftarrow \mathcal{D}]$, the probability that v is sampled from $\mathcal{D}$, can be lower bounded within an arbitrarily good precision in **AM**.

Lemma 4.4 (Informal). *If a language* L *can be reduced to a probability-verifiable sampling problem* S*, then* $\mathsf{L} \in \mathbf{AM} \cap \mathbf{coAM}$.

Lemma 4.4 is a generalization of [BB15]. Assume language L can be reduced to sampling problem S. The input of S is interpreted as the description of a distribution, let $\mathcal{P}_{\mathsf{pd}}$ denote the distribution specified by input pd.

Let $\mathcal{R}$ be the reduction from L to sampling problem S. On each input x, an execution $\mathcal{R}^{\mathsf{S}}(x)$ is determined by the random tape of reduction $\mathcal{R}$, denoted by r, and the answers to the reduction's queries to S. The *transcript* is defined as $\sigma = (r, \mathsf{pd}_1, v_1, \ldots, \mathsf{pd}_T, v_T)$ where pd_t is the t-th query to S and v_t is the corresponding response. Note that $r, v_1, \ldots, v_T$ determine the execution, since pd_t is determined by $r, v_1, \ldots, v_{t-1}$. Then

$$\Pr[\mathcal{R}^{\mathsf{S}}(x) \text{ accepts}] = \sum_{\substack{\sigma:\text{accepting transcript} \\ \text{of } \mathcal{R}^{\mathsf{S}}(x)}} \Pr[\sigma] = \sum_{\substack{\sigma:\text{accepting transcript} \\ \text{of } \mathcal{R}^{\mathsf{S}}(x)}} \Pr[r] \cdot \mathcal{P}_{\mathsf{pd}_1}(v_1) \cdot \ldots \cdot \mathcal{P}_{\mathsf{pd}_T}(v_T). \tag{1}$$

For simplicity, assume for now that there is an efficient algorithm that computes the probability $\mathcal{P}_{\mathsf{pd}}(v)$ given pd and value v. This property is stronger than probability-verifiability. Then the probability that $\mathcal{R}^{\mathsf{S}}(x)$ accepts, which equals a sum (Eq. (1)) where each term can be efficient computed, can be lower bounded using the set lower bound protocol of Goldwasser and Sipser [GS86], so $\mathsf{L} \in \mathbf{AM}$. Symmetrically, $\mathsf{L} \in \mathbf{coAM}$. The proof of Lemma 4.4 shows the same result from the weaker condition that S is probability-verifiable.

There is one last step missing between Lemmas 4.3 and 4.4: *Is discrete Gaussian sampling probability-verifiable? What is the smallest factor γ such that discrete Gaussian sampling with "width" $\leq \gamma\lambda_n$ is probability-verifiable?* Lemma 4.5 answers this question, and it connects the hardness of discrete Gaussian sampling with the hardness of $\mathsf{gapSIVP}$.

Lemma 4.5 (Informal). *Assume $\mathsf{gapSIVP}_\gamma$ is in* **SZK**. *There exists a real valued function $s(\mathbf{B}) \in [\lambda_n, \tilde{O}(\gamma) \cdot \lambda_n]$ such that given a lattice basis $\mathbf{B}$, discrete Gaussian sampling over lattice $\mathcal{L}(\mathbf{B})$ with "width" $s(\mathbf{B})$ is probability-verifiable.*

Lemma 4.5 has an easier proof assuming the stronger condition that $\mathsf{gapSIVP}_\gamma$ is in **P**. If there were some deterministic polynomial time algorithm solving $\mathsf{gapSIVP}_\gamma$, there would exist $s(\mathbf{B}) \in [\lambda_n(\mathbf{B}), \gamma\lambda_n(\mathbf{B})]$ that can be efficiently computed by binary search. As $s(\mathbf{B}) \geq \lambda_n(\mathbf{B})$, the verifier can ask the prover to provide a set of n linearly independent vectors $\mathbf{w}_1, \ldots, \mathbf{w}_n$ whose length is no longer than $s(\mathbf{B})$. Given the lattice basis $\mathbf{B}$ and a set of short linearly independent vectors, there exists an efficient algorithm that samples from the discrete Gaussian with the desired parameter [BLP+13]. When the verifier can sample from a distribution, he can lower bound the probability of each value using the set lower bound protocol [GS86].

This informal proof assumes $\mathsf{gapSIVP}_\gamma \in \mathbf{P}$ in order to compute a function $s(\mathbf{B})$ that $s(\mathbf{B}) \approx \lambda_n(\mathbf{B})$. As the verifier only needs to compute such a function $s(\mathbf{B})$ in an **AM** protocol, this assumption can be weakened to $\mathsf{gapSIVP}_\gamma \in \mathbf{SZK}$, by combining with Lemma 3.1.

Lemma 3.1 (Informal). *Assume $\mathsf{gapSIVP}_\gamma$ is in* **SZK**. *There exists a real valued function $s(\mathbf{B}) \in [\lambda_n, \tilde{O}(\gamma) \cdot \lambda_n]$ that can be efficiently computed in Arthur-Merlin protocol.*

The proof technique of Lemma 3.1 crucially relies on the fact that $\mathsf{gapSIVP}_\gamma \in \mathbf{SZK}$. As a result, we can hardly make use of previous results such as $\mathsf{gapSIVP}_{\sqrt{n/\log n}} \in \mathbf{NP} \cap \mathbf{coAM}$ [GMR04].

1.2 Related Works

Prior work exploring the problem of basing cryptography on worst-case **NP**-hardness has obtained several negative results for black-box reduction. Bras-

sard [Bra79] first showed that one-way permutations cannot be based on **NP**-hardness. Goldreich and Goldwasser [GG98] showed that public-key encryption schemes satisfying certain very specific properties cannot be based on **NP**-hardness. The required properties include the ability to certify an invalid key.

Work of Akavia, Goldreich, Goldwasser and Moshkovitz [AGGM06] and Bogdanov and Brzuska [BB15] showed that a special class of one-way functions called *size-verifiable one-way functions* cannot be based on **NP**-hardness. A size-verifiable one-way function is one in which the size of the set of pre-images can be efficiently approximated via an **AM** protocol.

Bogdanov and Lee [BL13] showed that homomorphic encryption schemes satisfying a special property cannot be based on **NP**-hardness. The required property is that the homomorphic evaluation produces a ciphertext whose distribution is statistically close to that of a fresh encrypted ciphertext.

Recently, Liu and Vaikuntanathan [LV16] showed that single-server private information retrieval (PIR) schemes cannot be based on **NP**-hardness.

Several works have also obtained a separation results for restricted types of reductions, most notably non-adaptive reductions which make all oracle queries simultaneously. The work of Feigenbaum and Fortnow [FF91], subsequently strengthened by Bogdanov and Trevisan [BT06], showed that there cannot be a *non-adaptive* reduction from SAT to the average-case hardness of any problem in **NP**, unless the polynomial hierarchy collapses.

On basing lattice problems on **NP**-hardness, the work of Goldreich and Goldwasser [GG00], subsequently strengthened by Micciancio and Vadhan [MV03], showed that $\mathsf{gapSVP}_{\sqrt{n/\log n}}$ and $\mathsf{gapCVP}_{\sqrt{n/\log n}}$ are both contained in $\mathbf{NP} \cap \mathbf{SZK}$. The shortest vector problem (SVP) and the closest vector problem (CVP), roughly speaking, is the problem of finding the shortest non-zero vector in a lattice or finding the lattice vector that is closest to a given point. The corresponding gap problem $\mathsf{gapSVP}_\gamma, \mathsf{gapCVP}_\gamma$ is to estimate within a factor of $\gamma(n)$ the length of the shortest non-zero vector or the distance to the closest lattice vector from a given point. The problem gapSVP is connected to $\mathsf{gapSIVP}$ via so-called "transference theorems" for lattices [Ban93]. Aharonov and Regev [AR04] explored a slightly looser approximation factor and showed that $\mathsf{gapSVP}_{\sqrt{n}}$ and $\mathsf{gapCVP}_{\sqrt{n}}$ are both contained in $\mathbf{NP} \cap \mathbf{coNP}$.

In prior work on the gap version of the SIVP problem, Guruswami, Micciancio and Regev [GMR04] showed that $\mathsf{gapSIVP}_{\sqrt{n/\log n}} \in \mathbf{NP} \cap \mathbf{coAM}$. Peikert and Vaikuntanathan [PV08] showed that $\mathsf{gapSIVP}_\gamma \in \mathbf{SZK}$ for any $\gamma(n) = \omega(\sqrt{n \log n})$. In contrast to these results for promise problems, our work explores the approximate SIVP problem. With an approximation factor $\gamma(n) = \tilde{O}(n)$, this search problem is the basis of lattice-based collision resistant hash function (CRHF) constructions [Ajt96, MR04]. In particular, Micciancio and Regev constructed CRHF from the worst-case hardness of $\mathsf{SIVP}_{\gamma(n)}$ for any $\gamma(n) = \omega(n \log n)$ [MR04]. We separate SIVP_γ from **NP**-hardness for the same approximation factor.

2 Preliminaries

Lattice A lattice in $\mathbb{R}^n$ is an additive subgroup of $\mathbb{R}^n$

$$\left\{\sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z} \text{ for } 1 \le i \le n\right\}$$

generated by n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^n$. The set of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ is called a *basis* for the lattice. A basis can be represented by matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$ whose columns are the basis vectors. The lattice generated by the columns of $\mathbf{B}$ is denoted by $\mathcal{L}(\mathbf{B})$.

$$\mathcal{L}(\mathbf{B}) = \{\mathbf{Bx} : \mathbf{x} \in \mathbb{N}^n\}.$$

The *i-th successive minimum* of a lattice $\mathcal{L}$, denoted by $\lambda_i(\mathcal{L})$, is defined as the minimum length that $\mathcal{L}$ contains i linearly independent vectors of length at most $\lambda_i(\mathcal{L})$. Formally,

$$\lambda_i(\mathcal{L}) := \min\{r : \dim(\mathcal{L} \cap r\mathcal{B}) \ge i\},$$

where $r\mathcal{B}$ is the radius r ball centered at the origin defined as $r\mathcal{B} := \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_2 \le r\}$. We abuse notations and write $\lambda_i(\mathbf{B})$ instead of $\lambda_i(\mathcal{L}(\mathbf{B}))$.

Shortest Independent Vectors Problem (SIVP). SIVP is a computational problem. Given a basis $\mathbf{B}$ of an n-dimensional lattice, find a set of n linearly independent vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathcal{L}(\mathbf{B})$ such that $\max_i \|\mathbf{v}_i\|$ is minimized, i.e., $\|\mathbf{v}_i\| \le \lambda_n(\mathbf{B})$ for all $1 \le i \le n$.

SIVP_γ is the approximation version of SIVP with factor λ. Given a basis $\mathbf{B}$ of an n-dimensional lattice, find a set of n linearly independent vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{v}_i\| \le \gamma(n) \cdot \lambda_n(\mathbf{B})$ for all $1 \le i \le n$. The approximation factor γ is typical a polynomial in n.

$\mathsf{gapSIVP}_\gamma$ is the decision version of SIVP_γ. An input to $\mathsf{gapSIVP}_\gamma$ is a basis $\mathbf{B}$ of a n-dimensional lattice and a scalar s. It is a YES instance if $\lambda_n(\mathbf{B}) \le s$, and is a NO instance if $\lambda_n(\mathbf{B}) \ge \gamma(n) \cdot s$.

Discrete Gaussian. For any vector $\mathbf{c}$ and any $s > 0$, let

$$\rho_{\mathbf{c},s}(\mathbf{v}) = e^{-\pi\|\mathbf{v}-\mathbf{c}\|_2^2/s^2}$$

be a Gaussian function with mean $\mathbf{c}$ and width s. Functions are extends to sets in usual way, $\rho_{\mathbf{c},s}(\mathcal{L}) = \sum_{\mathbf{v}\in\mathcal{L}} \rho_{\mathbf{c},s}(\mathbf{v})$. The discrete Gaussian distribution over lattice $\mathcal{L}$ with mean $\mathbf{c}$ and width s, denoted by $\mathcal{N}_{\mathcal{L},\mathbf{c},s}$, is defined by

$$\forall \mathbf{v} \in \mathcal{L},\ \mathcal{N}_{\mathcal{L},\mathbf{c},s}(\mathbf{v}) = \frac{\rho_{\mathbf{c},s}(\mathbf{v})}{\rho_{\mathbf{c},s}(\mathcal{L})}.$$

In this work, most discrete Gaussian distributions considered are centered at the origin. Let $\rho_s, \mathcal{N}_{\mathcal{L},s}$ denote $\rho_{\mathbf{0},s}, \mathcal{N}_{\mathcal{L},\mathbf{0},s}$ respectively.

Lemma 2.1 (Lemma 1.4 in [Ban93]). *For each $a \geq 1$, for any n-dimensional lattice $\mathcal{L}$, $\rho_{as}(\mathcal{L}) \leq a^n \rho_s(\mathcal{L})$*

Lemma 2.2 (Lemma 1.5 in [Ban93]). *For any $c > 1/\sqrt{2\pi}$, n-dimensional lattice $\mathcal{L}$*

$$\rho_s(\mathcal{L} \setminus cs\sqrt{n}\mathcal{B}) < C^n \cdot \rho_s(\mathcal{L}) \tag{2}$$

where $C = c\sqrt{2\pi e} \cdot e^{-\pi c^2}$.

Sampling Problems. Besides computational problems and decision problems, we define *sampling problems*. The input of a sampling problem specifies a distribution, let $\mathcal{P}_{\mathsf{pd}}$ denote the distribution specified by input pd. The goal is to sample from the distribution $\mathcal{P}_{\mathsf{pd}}$. A probabilistic polynomial-time algorithm $\mathcal{S}$ perfectly solves the sampling problem if for any input pd

$$\forall v, \Pr[\mathcal{S}(\mathsf{pd}) \to v] = \mathcal{P}_{\mathsf{pd}}(v).$$

The probability is over the random input tape of $\mathcal{S}$. In a more practical definition, $\mathcal{S}$ solves the sampling problem if the output distribution of $\mathcal{S}(\mathsf{pd})$ is close to $\mathcal{P}_{\mathsf{pd}}$, i.e.

$$\Delta_{\mathrm{sd}}(\mathcal{S}(\mathsf{pd}, 1^\ell), \mathcal{P}_{\mathsf{pd}}) \leq \frac{1}{\ell}$$

where Δ_{sd} denotes the statistical distance.

For example, in this work, discrete Gaussian is considered as a sampling problem. For any function $s(\cdot)$ mapping lattice bases to positive real numbers, define sampling problem DGS_s. The input of DGS_s is a lattice basis $\mathbf{B}$. The target output distribution $\mathcal{P}_{\mathbf{B}}$ is the discrete Gaussian distribution $\mathcal{N}_{\mathcal{L}(\mathbf{B}),s(\mathbf{B})}$, where each vector $v \in \mathcal{L}(\mathbf{B})$ is sampled with probability

$$\mathcal{P}_{\mathbf{B}}(\mathbf{v}) = \mathcal{N}_{\mathcal{L}(\mathbf{B}),s(\mathbf{B})}(\mathbf{v}) = \frac{\rho_{s(\mathbf{B})}(\mathbf{v})}{\rho_{s(\mathbf{B})}(\mathcal{L}(\mathbf{B}))}.$$

Probability-Verifiable. A sampling problem is *probability-verifiable* if there exists an **AM** protocol to lower bound $\mathcal{P}_{\mathsf{pd}}(v)$ for any pd and v. More precisely, there exists a family of error function $\{\eta_{\mathsf{pd},m}\}$ such that for any pd, m, the error function $\eta_{\mathsf{pd},m} : \{0,1\}^* \to [0, +\infty)$ satisfies $\sum_v \eta_{\mathsf{pd},m}(v) \leq \frac{1}{m}$, and the promise problem

- YES instance: $(\mathsf{pd}, v, \hat{p}, 1^m)$ such that $\hat{p} = \mathcal{P}_{\mathsf{pd}}(v)$
- NO instance: $(\mathsf{pd}, v, \hat{p}, 1^m)$ such that $\hat{p} \geq \mathcal{P}_{\mathsf{pd}}(v) + \eta_{\mathsf{pd},m}(v)$

is in **AM**.

Sampling Oracles. In order to formalize the (probabilistic) Turing reduction to a sampling problem, we also define *sampling oracles*, which is a generalization of traditional oracles studied by complexity theorists. Let $\mathcal{S}$ be a sampling oracle for a fixed sampling problem. $\mathcal{S}$ can be queried on any valid pd; upon query

pd, sampling oracle $\mathcal{S}(\mathsf{pd})$ would always output a fresh sample from distribution $\mathcal{P}_{\mathsf{pd}}$. E.g. if the sampling oracle $\mathcal{S}$ is queried for the same pd multiple times, it would output i.i.d. samples from distribution $\mathcal{P}_{\mathsf{pd}}$.

A probabilistic Turing reduction from a language L to a sampling problem S is a probabilistic poly-time oracle Turing machine $\mathcal{R}$, such that $\mathcal{R}$ can solve L given a sampling oracle of S in the sense that

$$x \in \mathsf{L} \implies \mathcal{R}^{\mathcal{S}}(x) \to 1 \text{ w.p. } \geq 2/3,$$
$$x \notin \mathsf{L} \implies \mathcal{R}^{\mathcal{S}}(x) \to 1 \text{ w.p. } \leq 1/3.$$

If such a reduction exists, we say L can be reduced to sampling problem S, denoted by $\mathsf{L} \in \mathbf{BPP}^{\mathsf{S}}$.

Similarly, a computational problem or a search problem can be reduced to a sampling problem S if they can be efficiently solved given the sampling oracle of S.

$\mathbb{R}$-TFAM *and* **$\mathbb{R}_\eta$-TFAM** The complexity class $\mathbb{R}$-**TFAM** is introduced by Mahmoody and Xiao [MX10]. Informally, it's consist of real-valued functions that can be efficiently computed in **AM**. A function $f : \{0,1\}^* \to \mathbb{R}$ is in $\mathbb{R}$-**TFAM** if the following promise problem is in **AM**:

- YES instance: $(x, f(x), 1^m)$.
- NO instance: $(x, y, 1^m)$ such that $|y - f(x)| > \frac{1}{m}$.

The definition of $\mathbb{R}$-**TFAM** emphasize on the absolute error. The complexity class $\mathbb{R}_\eta$-**TFAM** is defined to capture those functions that can be efficiently computed in **AM** with small relative error. A function $g : \{0,1\}^* \to \mathbb{R}^+$ is in $\mathbb{R}_\eta$-**TFAM** if the following promise problem is in **AM**:

- YES instance: $(x, g(x), 1^m)$.
- NO instance: $(x, y, 1^m)$ such that $|y - g(x)| > \frac{1}{m} \cdot g(x)$.

It follows directly from the definitions that $g \in \mathbb{R}_\eta - \mathbf{TFAM}$ if and only if $\log g \in \mathbb{R} - \mathbf{TFAM}$ for any function $g : \{0,1\}^* \to \mathbb{R}^+$.

Statistical Zero Knowledge. Statistical zero knowledge (**SZK**) is the class of decision problems that can be verified by a statistical zero-knowledge proof protocol. *Entropy Difference* (ED) is a complete problem for **SZK** [GV99], which is defined as the following: Given two polynomial-size circuits, C and D, let $\mathcal{C}$ and $\mathcal{D}$ be the distributions of their respective outputs when C, D are fed with uniform random inputs. The problem is to distinguish between

- YES instance: (C, D) such that $H(\mathcal{C}) - H(\mathcal{D}) \geq 1$;
- NO instance: (C, D) such that $H(\mathcal{C}) - H(\mathcal{D}) \leq -1$.

Where H is the Shannon entropy. Moreover, the mapping $H : C \mapsto H(\mathcal{C})$ is in $\mathbb{R}$-**TFAM**.

3 Gap Problems

The lattice problem gapSIVP is essentially estimating $\lambda_n(\mathbf{B})$ given a lattice basis $\mathbf{B}$. This definition can be generalized to any real valued functions. Define the gap problem of function $f : \{0,1\}^* \to \mathbb{R}^+$ with gap $\gamma : \{0,1\}^* \to [1, +\infty)$, denoted by $\mathsf{gap}f_\gamma$, as the promise problem

- YES instance: (x, y) such that $y \leq f(x)$;
- NO instance: (x, y) such that $y > \gamma(x) \cdot f(x)$.

In this work, estimating $\lambda_n(\mathbf{B})$ is of critical importance. Its gap problem, $\mathsf{gapSIVP}_\gamma$, alone is not sufficient for the proof. Instead, a stronger form of approximation is defined. Say $g : \{0,1\}^* \to \mathbb{R}^+$ is an approximation of function f within factor γ if $f(x) \leq g(x) \leq \gamma(x) \cdot f(x)$ for all x. Clearly, computing g is a harder problem than $\mathsf{gap}f_\gamma$, in the sense that there is a trivial reduction from $\mathsf{gap}f_\gamma$ to computing g.

The following Lemma shows a reduction in the other direction: if $\mathsf{gap}f_\gamma$ is in **SZK**, then there exists an approximation of f within almost the same factor, which can be computed in **AM**.

Lemma 3.1. *For any real valued function $f : \{0,1\}^* \to \mathbb{R}^+$ and any gap $\gamma : \{0,1\}^* \to [1, +\infty)$ that $\log \gamma(x) \leq \mathrm{poly}(|x|)$, if $\mathsf{gap}f_\gamma \in$ **SZK**, then for any constant $\mu > 1$, there exists $g : \{0,1\}^* \to \mathbb{R}^+$ such that $\forall x, g(x) \in [f(x), \mu\gamma(x)f(x)]$ and g is in $\mathbb{R}_\eta$-**TFAM**.*

Lemma 3.1 can be combined with previous results about gapSIVP. Peikert and Vaikuntanathan [PV08] showed that $\mathsf{gapSIVP}_\gamma \in$ **NISZK** $\subseteq$ **SZK** for any $\gamma = \omega(\sqrt{n \log n})$. Thus there exists an approximation of λ_n within a factor $\tilde{O}(\sqrt{n})$ that can be computed in **AM**.

Corollary 3.2. *For any $\gamma(n) = \omega(\sqrt{n \log n})$, there exists a function g maps lattice bases to real numbers such that $g \in \mathbb{R}_\eta -$ **TFAM** and $\lambda_n(\mathbf{B}) \leq g(\mathbf{B}) < \gamma(n) \cdot \lambda_n(\mathbf{B})$.*

Proof (Lemma 3.1). Entropy Difference (ED) is a complete problem for **SZK**, so $\mathsf{gap}f_\gamma \in$ **SZK** implies the existence of a reduction $(x, y) \mapsto (C_{x,y}, D_{x,y})$ that maps input x together with a real number y to random circuits $C_{x,y}, D_{x,y}$. Let $\mathcal{C}_{x,y}$ and $\mathcal{D}_{x,y}$ be the output distributions of $C_{x,y}, D_{x,y}$. The reduction from $\mathsf{gap}f_\gamma$ to ED satisfies the following properties:

- There is an efficient deterministic algorithm computing $C_{x,y}, D_{x,y}$ given input (x, y).
- $H(\mathcal{C}_{x,y}) - H(\mathcal{D}_{x,y}) > 2$ for any x, y that $y \leq f(x)$.
- $H(\mathcal{C}_{x,y}) - H(\mathcal{D}_{x,y}) < -1$ for any x, y that $y > \gamma(x) \cdot f(x)$.

Define the clamp function

$$\mathrm{clamp}(y) := \begin{cases} 1, & \text{if } y \geq 1; \\ y, & \text{if } y \in (0, 1); \\ 0, & \text{if } y \leq 0. \end{cases}$$

For any fixed constant $\mu > 1$, define

$$g(x) = \exp\left(\ln\mu \cdot \sum_{i=0}^{+\infty} \operatorname{clamp}(H(\mathcal{C}_{x,\mu^i}) - H(\mathcal{D}_{x,\mu^i})) + \ln\mu \cdot \sum_{i=1}^{+\infty} \left(\operatorname{clamp}(H(\mathcal{C}_{x,\mu^{-i}}) - H(\mathcal{D}_{x,\mu^{-i}})) - 1\right)\right).$$

As $\operatorname{clamp}(H(\mathcal{C}_{x,y}) - H(\mathcal{D}_{x,y})) = 1$ for $y \le f(x)$,

$$g(x) \ge \exp\bigl(\ln\mu \cdot \lceil \log_\mu(f(x)) \rceil\bigr) \ge f(x).$$

As $\operatorname{clamp}(H(\mathcal{C}_{x,y}) - H(\mathcal{D}_{x,y})) = 0$ for $y > \gamma(x) \cdot f(x)$,

$$g(x) \le \exp\bigl(\ln\mu \cdot \lceil \log_\mu(\gamma(x) \cdot f(x)) \rceil\bigr) \le \mu\gamma(x) \cdot f(x).$$

In order to complete the proof, we show that g is in $\mathbb{R}_\eta$-**TFAM**. For any input $x, \hat{g}$, the prover can prove $\hat{g} \approx g(x)$ if $\hat{g} = g(x)$.

Consider the following protocol, $\varepsilon = 1/\operatorname{poly}(m, \ln\gamma)$ will be fixed later.

On any input x, define $d_i = H(\mathcal{C}_{x,\mu^i}) - H(\mathcal{D}_{x,\mu^i})$. And the honest prover should send $\hat{d}_i = d_i$. The prover have to prove that $d_i - \varepsilon < \hat{d}_i < d_i + \varepsilon$. For $\mu^i \le f(x)$, $\hat{d}_i \ge d_i - \varepsilon \ge 1$, then $\operatorname{clamp}(\hat{d}_i) = 1 = \operatorname{clamp}(d_i)$. For $\mu^i \ge \mu\gamma(x)f(x)$, $\hat{d}_i \le d_i + \varepsilon \le 0$, then $\operatorname{clamp}(\hat{d}_i) = 0 = \operatorname{clamp}(d_i)$. For $f(x) < \mu^i < \mu\gamma(x)f(x)$, $|\operatorname{clamp}(\hat{d}_i) - \operatorname{clamp}(d_i)| \le |\hat{d}_i - d_i| < \varepsilon$.

AM "protocol" on input $(x, \hat{g})$

P: Send $\ldots, \hat{d}_{-1}, \hat{d}_0, \hat{d}_1, \hat{d}_2, \ldots$ such that $\log_\mu \hat{g} = \sum_{i=0}^{\infty} \operatorname{clamp}(\hat{d}_i) + \sum_{i=1}^{\infty} (\operatorname{clamp}(\hat{d}_{-i}) - 1)$

P,V: For each $i \in \mathbb{Z}$, convince the verifier that $\hat{d}_i - \varepsilon < H(\mathcal{C}_{x,\mu^i}) - H(\mathcal{D}_{x,\mu^i}) < \hat{d}_i + \varepsilon$

Thus

$$\begin{aligned}
\left|\frac{\ln\hat{g} - \ln g(x)}{\ln\mu}\right| &\le \sum_{i\in\mathbb{Z}} \bigl|\operatorname{clamp}(\hat{d}_i) - \operatorname{clamp}(d_i)\bigr| \\
&= \sum_{f(x)<\mu^i<\mu\gamma(x)f(x)} \bigl|\operatorname{clamp}(\hat{d}_i) - \operatorname{clamp}(d_i)\bigr| \\
&< \lceil \log_\mu(\mu\gamma(x)) \rceil \varepsilon \\
&< \frac{\ln\gamma(x) + 2}{\ln\mu}\varepsilon.
\end{aligned}$$

If ε is sufficiently small, $\hat{g}$ would be close to $g(x)$. To ensure $|\hat{g} - g(x)| \le \frac{1}{m} g(x)$, it is sufficient to set $\varepsilon = O(\frac{1}{m(\ln\gamma(x)+2)})$.

The above "protocol" is not a real protocol, as it requires the prover to send an infinite sequence to the verifier. To compress the proof, the prover need a succinct interactive proof that $d_j > 1$ for all $j \le i_L$ and $d_j < 0$ for all $j \ge i_H$.

For an index i, if the prover can convince the verifier that $d_i = H(\mathcal{C}_{x,\mu^i}) - H(\mathcal{D}_{x,\mu^i}) < 2$, the verifier also learns that $\mu^i > g(x)$, thus for any $j \ge i +$

$\lceil \log_\mu \gamma(x) \rceil$, $\mu^j > \gamma(x)g(x)$ and $d_j \leq -1$. Similarly, if the prover can convince the verifier that $d_i = H(\mathcal{C}_{x,\mu^i}) - H(\mathcal{D}_{x,\mu^i}) > -1$, the verifier also knows that $d_j \geq 2$ for any $j \leq i - \lceil \log_\mu \gamma(x) \rceil$.

Thus the real **AM** protocol that proves $\hat{g} \in (g(x) - \frac{1}{m}, g(x) + \frac{1}{m})$ is the following: □

AM protocol on input $(x, \hat{g}, 1^m)$

P: Send $\hat{d}_{i_L}, \hat{d}_{i_L+1}, \ldots, \hat{d}_{i_H-1}, \hat{d}_{i_H}$ such that
- $\log_\mu \hat{g} = i_L + \sum_{i=i_L}^{i_H} \text{clamp}(\hat{d}_i)$
- $i_H = i_L + 2\lceil \log_\mu \gamma(x) \rceil$
- $\hat{d}_{i_L + \lceil \log_\mu \gamma(x) \rceil} > 0$
- $\hat{d}_{i_L + \lceil \log_\mu \gamma(x) \rceil + 1} < 1$

P,V: For each $i \in \mathbb{Z}$, convince the verifier that $\hat{d}_i - \varepsilon < H(\mathcal{C}_{x,\mu^i}) - H(\mathcal{D}_{x,\mu^i}) < \hat{d}_i + \varepsilon$ for $\varepsilon = O(\frac{1}{m(\ln \gamma(x)+2)})$.

---

4 Search SIVP and NP-Hardness

Theorem 4.1. *For any factor* $\gamma : \mathbb{N} \to \mathbb{R}$*, if* $\mathsf{gapSIVP}_\gamma \in$ **SZK** *and there exists a probabilistic polynomial-time adaptive reduction from a language* L *to* $\mathsf{SIVP}_{\sqrt{n \ln n} \cdot \gamma}$*, then* $\mathsf{L} \in$ **AM** $\cap$ **coAM**.

The smallest factor γ we knows that makes problem $\mathsf{gapSIVP}_\gamma$ be in **SZK** comes from [PV08]: for any factor $\gamma(n) = \omega(\sqrt{n \log n})$, problem $\mathsf{gapSIVP}_\gamma$ is in **SZK**.

Corollary 4.2. *For any factor* $\gamma(n) = \omega(n \log n)$*, if there exists a probabilistic polynomial-time adaptive reduction from a language* L *to* SIVP_γ*, then* $\mathsf{L} \in$ **AM** $\cap$ **coAM**.

The proof of Theorem 4.1 is the combination of Lemmas 4.3, 4.4 and 4.5. Problem $\mathsf{gapSIVP}_\gamma$ is in **SZK** and there is a reduction from language L to search problem $\mathsf{SIVP}_{\sqrt{n \ln n} \cdot \gamma}$. Lemma 4.3 shows that there is another reduction from L to sampling problem DGS_s for any s satisfying

$$s(\mathbf{B}) \in [\lambda_n(\mathbf{B}), \sqrt{\ln n} \cdot \gamma \lambda_n(\mathbf{B})]. \tag{3}$$

Lemma 4.5 shows that there exists a function s satisfying (3) such that the sampling problem DGS_s is probability-verifiable. Therefore, there exists a reduction from L to a probability-verifiable sampling problem. Finally, Lemma 4.4 shows that such a language L must live in **AM** $\cap$ **coAM**.

Lemma 4.3. *Let $s(\cdot)$ be a function mapping lattice bases to real numbers, such that $\forall \mathbf{B}, \lambda_n(\mathbf{B}) \leq s(\mathbf{B}) \leq \frac{\gamma}{\sqrt{n}}\lambda_n(\mathbf{B})$. Then there exists a probabilistic Turing reduction from SIVP_γ to DGS_s.*

Lemma 4.4. *If there exists a probabilistic Turing reduction from a promise problem $\mathsf{L} = (\mathsf{L}_Y, \mathsf{L}_N)$ to probability-verifiable sampling problems, then $\mathsf{L} \in \mathbf{AM} \cap \mathbf{coAM}$.*

Lemma 4.5. *For any factor $\gamma : \mathbb{N} \to \mathbb{R}$, if $\mathsf{gapSIVP}_{\gamma(n)/\sqrt{\ln n}} \in \mathbf{SZK}$, then there exists a function $s(\cdot)$ mapping lattice bases to real numbers, such that $\forall \mathbf{B}, s(\mathbf{B}) \in [\lambda_n(\mathbf{B}), \gamma(n) \cdot \lambda_n(\mathbf{B})]$ and DGS_s is probability-verifiable.*

By combining Lemmas 4.4, 4.5 and [PV08], we can also show that discrete Gaussian sampling with width $\tilde{O}(\sqrt{n}) \cdot \lambda_n$ is not **NP**-hard unless the polynomial hierarchy collapses.

Theorem 4.6. *If there exists a probabilistic Turing reduction from a promise problem L to DGS_s for $s(\mathbf{B}) = \omega(\sqrt{n}\log n) \cdot \lambda_n(\mathbf{B})$, then $\mathsf{L} \in \mathbf{AM} \cap \mathbf{coAM}$.*

4.1 From Search SIVP to Discrete Gaussian Sampling

This section proves Lemma 4.3, which is essentially Lemma 3.17 in Regev's work [Reg09]. Informally speaking, Regev shows a reduction from SIVP_γ to $\mathsf{DGS}_{\gamma/\sqrt{n}}$ for $\gamma = \Omega(\sqrt{n \log n})$; Lemma 4.3 uses similar technique to construct a reduction from SIVP_γ to $\mathsf{DGS}_{\gamma/\sqrt{n}}$ for $\gamma = \Omega(\sqrt{n})$.

The reduction from SIVP_γ to discrete Gaussian sampling is straightforward: Sample n^2 times from discrete Gaussian distribution of width $s \in [\lambda_n, \frac{\gamma}{\sqrt{n}}\lambda_n]$. The sampled vectors contain n short, linearly independent vectors with probability exponentially close to 1.

In order to prove Lemma 4.3, we shows that if n^2 vectors are sampled from discrete Gaussian $\mathcal{N}_{\mathcal{L}(\mathbf{B}),s(\mathbf{B})}$, the following two "bad events" occurs with probability exponentially small.

- One of the sampled vectors is too long, its Euclidean norm is larger than $\gamma\lambda_n(\mathbf{B})$.
- The sampled vectors are not full rank.

Lemma 2.2 bounds the probability that an overlong vector is sampled from a discrete Gaussian distribution. Let the constant c in formula (2) equals 1,

$$\Pr_{\mathbf{v} \leftarrow \mathcal{N}_{\mathcal{L}(\mathbf{B}),s(\mathbf{B})}}\left[\|\mathbf{v}\| > \sqrt{n} \cdot s(\mathbf{B})\right] = \frac{\rho_s(\mathcal{L}(\mathbf{B}) \setminus s\sqrt{n}\mathcal{B})}{\rho_s(\mathcal{L}(\mathbf{B}))} < \left(\sqrt{2\pi e} \cdot e^{-\pi}\right)^n < 0.2^n.$$

As $\gamma(n) \cdot \lambda_n(\mathbf{B}) \geq \sqrt{n} \cdot s(\mathbf{B})$,

$$\Pr_{\mathbf{v} \leftarrow \mathcal{N}_{\mathcal{L}(\mathbf{B}),s(\mathbf{B})}}\left[\|\mathbf{v}\| > \gamma\lambda_n(\mathbf{B})\right] \leq \Pr_{\mathbf{v} \leftarrow \mathcal{N}_{\mathcal{L}(\mathbf{B}),s(\mathbf{B})}}\left[\|\mathbf{v}\| > \sqrt{n} \cdot s(\mathbf{B})\right] < 0.2^n,$$

which is exponentially small.

To prove that the n^2 sampled vectors span the whole space, we need a lower bound on the probability a newly sampled vector is linear independent from the previous ones. Lemma 4.7 shows such a lower bound, improves [Reg09, Lemma 3.15] by a factor of $\sqrt{\ln n}$ (the so-called smoothing parameter).

Lemma 4.7. *For any n-dimensional lattice $\mathcal{L}$, real number $s \geq \lambda_n(\mathcal{L})$ and for any proper linear subspace $\mathcal{V} \subsetneq \mathbb{R}^n$, the probability $\Pr_{\mathbf{v} \leftarrow \mathcal{N}_{\mathcal{L},s}}[\mathbf{v} \notin \mathcal{V}]$ is at least $1/20$.*

Proof. By the definition of successive minimum, there exists $\mathbf{u} \in \mathcal{L} \setminus \mathcal{V}$ such that $\|\mathbf{u}\| \leq \lambda_n(\mathcal{L})$. Let $\mathcal{L}'$ denote $\mathcal{L} \cap \mathcal{V}$. As $\mathcal{L}$ is closed under addition, $\mathcal{L}' + \mathbf{u}, \mathcal{L}' - \mathbf{u}$ are subsets of $\mathcal{L}$. Moreover, as $\mathcal{V}$ is closed under addition and $\mathbf{u} \notin \mathcal{V}$, the sets $\mathcal{L}' + \mathbf{u}, \mathcal{L}', \mathcal{L}' - \mathbf{u}$ are disjointed.

$$\begin{aligned}\Pr_{\mathbf{v} \leftarrow \mathcal{N}_{\mathcal{L},s}}[\mathbf{v} \in \mathcal{V}] &= \frac{\rho_s(\mathcal{L}')}{\rho_s(\mathcal{L})} \\ &\leq \frac{\rho_s(\mathcal{L}')}{\rho_s(\mathcal{L}' - \mathbf{u}) + \rho_s(\mathcal{L}') + \rho_s(\mathcal{L}' + \mathbf{u})} = \frac{\sum_{\mathbf{v} \in \mathcal{L}'} \rho_s(\mathbf{v})}{\sum_{\mathbf{v} \in \mathcal{L}'} (\rho_s(\mathbf{v} - \mathbf{u}) + \rho_s(\mathbf{v}) + \rho_s(\mathbf{v} + \mathbf{u}))}\end{aligned}$$

As $\|\mathbf{u}\| \leq \lambda_n(\mathcal{L}) \leq s$, for any vector $\mathbf{v}$

$$\begin{aligned}\rho_s(\mathbf{v} - \mathbf{u}) + \rho_s(\mathbf{v} + \mathbf{u}) &= e^{-\pi\|\mathbf{v}-\mathbf{u}\|^2/s^2} + e^{-\pi\|\mathbf{v}-\mathbf{u}\|^2/s^2} \\ &= (e^{-2\pi\langle \mathbf{u},\mathbf{v}\rangle/s^2} + e^{2\pi\langle \mathbf{u},\mathbf{v}\rangle/s^2}) e^{-\pi\|\mathbf{u}\|^2/s^2} e^{-\pi\|\mathbf{v}\|^2/s^2} \leq 2e^{-\pi}\rho_s(\mathbf{v})\end{aligned}$$

Thus

$$\Pr_{\mathbf{v} \leftarrow \mathcal{N}_{\mathcal{L},s}}[\mathbf{v} \in \mathcal{V}] \leq \frac{\sum_{\mathbf{v} \in \mathcal{L}'} \rho_s(\mathbf{v})}{\sum_{\mathbf{v} \in \mathcal{L}'} (1 + 2e^{-\pi/2^2})\rho_s(\mathbf{v})} = \frac{1}{1 + 2e^{-\pi}} \approx 0.92.$$

□

Assume k vectors has been sampled from $\mathcal{N}_{\mathcal{L}(\mathbf{B}),s(\mathbf{B})}$ and their dimension is strictly less than n. By Lemma 4.7, the next n sampled vectors contain a vector linearly independent from the first k with probability exponentially close to 1. By union bound, n^2 samples from $\mathcal{N}_{\mathcal{L}(\mathbf{B}),s(\mathbf{B})}$ contains n linearly independent vectors with probability exponentially close to 1.

4.2 Probability-Verifiable Sampling Problem and NP-hardness

This section proves Lemma 4.4, which is a generalization of [BB15], the proof techniques are similar.

Let $\mathcal{M}$ be the reduction from a promise problem $\mathsf{L} = (\mathsf{L}_Y, \mathsf{L}_N)$ to $\mathcal{S}$. For a given input x, we want to distinguish between $\Pr[\mathcal{M}^{\mathcal{S}}(x) \to 1] \geq 8/9$ and $\Pr[\mathcal{M}^{\mathcal{S}}(x) \to 1] \leq 1/9$ in **AM**. Notice that the randomness includes the random tape of $\mathcal{M}$ and the randomness $\mathcal{S}$ used to answer each query.

A transcript of an execution of $\mathcal{M}^{\mathcal{S}}(x)$ is an tuple $(r, \mathsf{pd}_1, v_1, \mathsf{pd}_2, v_2, \ldots, \mathsf{pd}_T, v_T)$ consists of the random tape of $\mathcal{M}$, all queries to $\mathcal{S}$ and the correlated answers. The transcript fully determined the execution $\mathcal{M}^{\mathcal{S}}(x)$, and

$$\begin{aligned}\Pr[\mathcal{M}^{\mathcal{S}}(x) \to 1] &= \sum_{\substack{\text{transcript } (r,\mathsf{pd}_1,v_1,\mathsf{pd}_2,v_2,\ldots,\mathsf{pd}_T,v_T) \\ \text{determines a execution where } \mathcal{M}^{\mathcal{S}}(x)\to 1}} \Pr[(r, \mathsf{pd}_1, v_1, \mathsf{pd}_2, v_2, \ldots, \mathsf{pd}_T, v_T)] \\ &= \sum_{\substack{\text{transcript } (r,\mathsf{pd}_1,v_1,\mathsf{pd}_2,v_2,\ldots,\mathsf{pd}_T,v_T) \\ \text{determines a execution where } \mathcal{M}^{\mathcal{S}}(x)\to 1}} \Pr[r] \prod_{t=1}^{T} \mathcal{P}_{\mathsf{pd}_t}(v_t).\end{aligned}$$

In the proof, we construct an **AM** protocol that estimates this sum.

Proof of Lemma 4.4. It's sufficient to show that $\mathsf{L} = (\mathsf{L}_Y, \mathsf{L}_N) \in \mathbf{AM}$. Then the same argument would shows $\bar{\mathsf{L}} = (\mathsf{L}_N, \mathsf{L}_Y) \in \mathbf{AM}$, which implies $\mathsf{L} \in \mathbf{coAM}$.

L can be efficiently reduced to a probability-verifiable sampling problem. Let $\mathcal{S}$ denote a correlated sampling oracle. The reduction is a probability polynomial-time oracle algorithm $\mathcal{M}$ such that

$$\begin{aligned} x \in \mathsf{L}_Y &\implies \Pr[\mathcal{M}^{\mathcal{S}}(x) \to 1] \geq \frac{8}{9}, \\ x \in \mathsf{L}_N &\implies \Pr[\mathcal{M}^{\mathcal{S}}(x) \to 1] \leq \frac{1}{9}. \end{aligned} \tag{4}$$

The probability is over the random tape of $\mathcal{M}$ and the randomness used by $\mathcal{S}$. Without loss of generality, assume there exists $T = \mathrm{poly}(n)$ that $\mathcal{M}$ uses T bits of randomness and makes T queries on any input $x \in \{0,1\}^n$.

Define a *transcript* of an execution $\mathcal{M}^{\mathcal{S}}(x)$ as a tuple $(r, \mathsf{pd}_1, v_1, \mathsf{pd}_2, v_2, \ldots, \mathsf{pd}_T, v_T)$ where $r \in \{0,1\}^T$ is the random tape of $\mathcal{M}$, pd_t is the t-th query to sampling oracle $\mathcal{S}$ and v_t is the t-th sample returned by $\mathcal{S}$. The length of v_t is bounded by some polynomial of n, let $\ell(n)$ be a polynomial that upper bound $|v_t|$.

Note that the input, the random tape and oracle's answers fully determine the reduction. Given the input and random tape, the reduction's first query is predictable; given the input, random tape and the oracle's previous answers, the reduction's next query is predictable. Therefore, we define a transcript $\sigma = (r, \mathsf{pd}_1, v_1, \mathsf{pd}_2, v_2, \ldots, \mathsf{pd}_T, v_T)$ to be *valid*, if it's potentially a transcript of an execution $\mathcal{M}^{\mathcal{S}}(x)$, i.e. if for all $1 \leq t \leq T$, pd_t would the t-th query in execution $\mathcal{M}^{\mathcal{S}}(x)$ when r is the random tape and $v_1, \ldots, v_{t-1}$ is the oracle's previous answers. By this definition, σ is a valid transcript doesn't implies v_t has non-zero probability under distribution pd_t. Let $C(x)$ denote the set of all valid transcripts of $\mathcal{M}^{\mathcal{S}}(x)$.

The transcript also determines the output of the reduction. Define a transcript σ to be *accepting*, if σ is valid and the corresponding execution $\mathcal{M}^{\mathcal{S}}(x)$ output 1. Let $C_1(x)$ denote the set of all accepting transcripts of $\mathcal{M}^{\mathcal{S}}(x)$.

Let $P_x(\sigma)$ denote the probability that σ is the transcript of $\mathcal{M}^{\mathcal{S}}(x)$ when the random tape is uniformly chosen and $\mathcal{S}$ is an ideal sampling oracle. Then by chain rule,

$$P_x(\sigma) = \frac{1}{2^T} \prod_{t=1}^{T} \mathcal{P}_{\mathsf{pd}_t}(v_t)$$

for any valid transcript $\sigma = (r, \mathsf{pd}_1, v_1, \mathsf{pd}_2, v_2, \ldots, \mathsf{pd}_T, v_T)$. For any input x, we know $C_1(x) \subseteq C(x)$,

$$\sum_{\sigma \in C(x)} P_x(\sigma) = 1, \qquad \sum_{\sigma \in C_1(x)} P_x(\sigma) = \Pr[\mathcal{M}^{\mathcal{S}}(x) \to 1]$$

by the definition of valid/accepting transcripts. Thus, by condition (4), to distinguish between $x \in \mathsf{L}_Y$ and $x \in \mathsf{L}_N$, it's sufficient to distinguish between $\sum_{\sigma \in C_1(x)} P_x(\sigma) \geq 8/9$ and $\sum_{\sigma \in C_1(x)} P_x(\sigma) \leq 1/9$.

Define $D(x)$ as the set of all tuple (σ, k) such that $\sigma = (r, \mathsf{pd}_1, v_1, \mathsf{pd}_2, v_2, \ldots, \mathsf{pd}_T, v_T) \in C_1(x)$, and k is an integer that

$$1 \leq k \leq K \cdot P_x(\sigma) = K \cdot \frac{1}{2^T} \prod_{t=1}^{T} \mathcal{P}_{\mathsf{pd}_t}(v_t)$$

where $K = 10 \cdot 2^T \cdot 2^{T(\ell+1)}$. Then the size of $D(x)$ is roughly $K \cdot \Pr[\mathcal{M}^{\mathcal{S}}(x) \to 1]$ if K is sufficiently large.

The sampling problem is probability-verifiable. By definition, there exists a family of error function $\{\eta_{\mathsf{pd},m}\}$ such that for any pd, m, the error function $\eta_{\mathsf{pd},m} : \{0,1\}^* \to [0, +\infty)$ satisfies $\sum_v \eta_{\mathsf{pd},m}(v) \leq 1$, and the promise problem

- YES instances: $(\mathsf{pd}, v, \hat{p}, 1^m)$ such that $\hat{p} = \mathcal{P}_{\mathsf{pd}}(v)$
- NO instances: $(\mathsf{pd}, v, \hat{p}, 1^m)$ such that $\hat{p} \geq \mathcal{P}_{\mathsf{pd}}(v) + \frac{1}{m}\eta_{\mathsf{pd},m}(v)$

is in **AM**. Let ProbLowerBound be the corresponding **AM** protocol.

Let set $D'(x)$ consist of all tuple (σ, k) such that $\sigma = (r, \mathsf{pd}_1, v_1, \mathsf{pd}_2, v_2, \ldots, \mathsf{pd}_T, v_T) \in C_1(x)$, and k is an integer that

$$1 \leq k \leq K \cdot \frac{1}{2^T} \prod_{t=1}^{T} \Big(\mathcal{P}_{\mathsf{pd}_t}(v_t) + \frac{1}{T} \eta_{\mathsf{pd}_t,T}(v_t) \Big).$$

Here $K = 10 \cdot 2^T \cdot 2^{T(\ell+1)}$ as in the definition of $D(x)$. By definition, $D(x) \subseteq D'(x)$.

Claim. The promise problem

- YES instances: (x, σ, k) such that $(\sigma, k) \in D(x)$
- NO instances: (x, σ, k) such that $(\sigma, k) \notin D'(x)$

is in **AM**.

Proof. TranscriptChecking is an **AM** protocol that solves this promise problem.

AM protocol TranscriptChecking on input $(x, \sigma = (r, \mathsf{pd}_1, v_1, \mathsf{pd}_2, v_2, \ldots, \mathsf{pd}_T, v_T), k)$

V: Check whether σ is a valid accepting transcript of $\mathcal{M}^{\mathcal{S}}(x)$; Reject if not
P: Send $\hat{p}_1, \ldots, \hat{p}_T$, an honest prover should send $\hat{p}_t = \mathcal{P}_{\mathsf{pd}_t}(v_t)$
P,V: Run protocol ProbLowerBound$(\mathsf{pd}_t, v_t, 1^{10T})$ for all $1 \leq t \leq T$, repeat polynomial many times in parallel and take majority so that the total error probability is exponentially small; Reject if either of these protocols reject.
V: Check whether $1 \leq k \leq K \cdot \frac{1}{2^T} \prod_{i=1}^{q} \hat{p}_i$; Reject if not

---

For $(\sigma, k) \in D(x)$, an honest prover could convince the verifier that to accept (x, σ, k).

Any prover, even if it's malicious, should send $\hat{p}_t$ such that $\hat{p}_t \leq \mathcal{P}_{\mathsf{pd}_t}(v_t) + \frac{1}{10T}\eta_{\mathsf{pd}_t, 10T}(v_t)$. Otherwise the prover will be caught in ProbLowerBound protocol with overwhelming probability. Thus no prover can make the verifier accept (x, σ, k) with high probability if $(\sigma, k) \notin D'(x)$. □

Claim. The size of $D(x)$ is at least $\frac{2}{3}K$ if $x \in \mathsf{L}_Y$.

Proof. $x \in \mathsf{L}_Y$ implies that $\Pr[\mathcal{M}^{\mathcal{S}}(x) \to 1] \geq \frac{8}{9}$. Thus

$$\begin{aligned}
|D(x)| &= \sum_{\sigma \in C_1(x)} \lfloor K \cdot P_x(\sigma) \rfloor \\
&\geq \sum_{\sigma \in C_1(x)} (K \cdot P_x(\sigma) - 1) \\
&= K \cdot \sum_{\sigma \in C_1(x)} P_x(\sigma) - |C_1(x)| \\
&\geq K \cdot \Pr[\mathcal{M}^{\mathcal{S}}(x) \to 1] - |C(x)| \\
&\geq \frac{8}{9}K - 2^T \cdot 2^{T(\ell+1)} \\
&= \frac{8}{9}K - \frac{1}{10}K \\
&\geq \frac{2}{3}K
\end{aligned}$$

□

Claim. $D'(x)$ has size at most $\frac{1}{3}K$ if $x \in \mathsf{L}_N$.

Proof. $x \in \mathsf{L}_N$ implies that $\Pr[\mathcal{M}^{\mathcal{S}}(x) \to 1] \leq \frac{1}{9}$.

$$
\begin{aligned}
|D'(x)| &= \sum_{\sigma=(r,\mathsf{pd}_1,v_1,\mathsf{pd}_2,v_2,\ldots,\mathsf{pd}_T,v_T)\in C_1(x)} \Big\lfloor K \cdot \frac{1}{2^T}\prod_{t=1}^{T}\Big(\mathcal{P}_{\mathsf{pd}_t}(v_t) + \frac{1}{10T}\eta_{\mathsf{pd}_t,10T}(v_t)\Big)\Big\rfloor \\
&\leq K \cdot \sum_{\sigma=(r,\mathsf{pd}_1,v_1,\mathsf{pd}_2,v_2,\ldots,\mathsf{pd}_T,v_T)\in C_1(x)} \frac{1}{2^T}\prod_{t=1}^{T}\Big(\mathcal{P}_{\mathsf{pd}_t}(v_t) + \frac{1}{10T}\eta_{\mathsf{pd}_t,10T}(v_t)\Big) \\
&= K \cdot \sum_{\sigma=(r,\mathsf{pd}_1,\ldots,v_T)\in C_1(x)} \left(\frac{1}{2^T}\prod_{t=1}^{T}\Big(\mathcal{P}_{\mathsf{pd}_t}(v_t) + \frac{1}{10T}\eta_{\mathsf{pd}_t,10T}(v_t)\Big) - \frac{1}{2^T}\prod_{t=1}^{T}\mathcal{P}_{\mathsf{pd}_t}(v_t)\right) \\
&\quad + K \cdot \sum_{\sigma=(r,\mathsf{pd}_1,v_1,\mathsf{pd}_2,v_2,\ldots,\mathsf{pd}_T,v_T)\in C_1(x)} \frac{1}{2^T}\prod_{t=1}^{T}\mathcal{P}_{\mathsf{pd}_t}(v_t) \\
&\leq K \cdot \sum_{\sigma=(r,\mathsf{pd}_1,\ldots,v_T)\in C(x)} \left(\frac{1}{2^T}\prod_{t=1}^{T}\Big(\mathcal{P}_{\mathsf{pd}_t}(v_t) + \frac{1}{10T}\eta_{\mathsf{pd}_t,10T}(v_t)\Big) - \frac{1}{2^T}\prod_{t=1}^{T}\mathcal{P}_{\mathsf{pd}_t}(v_t)\right) \\
&\quad + K \cdot \Pr[\mathcal{M}^{\mathcal{S}}(x) \to 1] \\
&\leq (e^{1/10}-1)K + \frac{1}{9}K \\
&\leq \frac{1}{3}K.
\end{aligned}
$$

The second to last inequality symbol relies on the following inequality,

$$
\begin{aligned}
&\sum_{\sigma=(r,\mathsf{pd}_1,v_1,\ldots,\mathsf{pd}_T,v_T)\in C(x)} \left(\frac{1}{2^T}\prod_{t=1}^{T}\Big(\mathcal{P}_{\mathsf{pd}_t}(v_t) + \frac{1}{10T}\eta_{\mathsf{pd}_t,10T}(v_t)\Big)\right) \\
&= \sum_{\substack{(r,\mathsf{pd}_1,v_1,\ldots,\mathsf{pd}_{T-1},v_{T-1},\mathsf{pd}_T) \\ \exists v_T\ (r,\mathsf{pd}_1,v_1,\ldots,\mathsf{pd}_T,v_T)\in C(x)}} \left(\frac{1}{2^T}\prod_{t=1}^{T-1}\Big(\mathcal{P}_{\mathsf{pd}_t}(v_t) + \frac{1}{10T}\eta_{\mathsf{pd}_t,10T}(v_t)\Big)\cdot \right. \\
&\qquad\qquad \left. \sum_{v}\Big(\mathcal{P}_{\mathsf{pd}_T}(v) + \frac{1}{10T}\eta_{\mathsf{pd}_T,10T}(v)\Big)\right) \\
&\leq \sum_{\substack{(r,\mathsf{pd}_1,v_1,\ldots,\mathsf{pd}_{T-1},v_{T-1}) \\ \exists \mathsf{pd}_T,v_T\ (r,\mathsf{pd}_1,\ldots,v_T)\in C(x)}} \left(\frac{1}{2^T}\prod_{t=1}^{T-1}\Big(\mathcal{P}_{\mathsf{pd}_t}(v_t) + \frac{1}{10T}\eta_{\mathsf{pd}_t,10T}(v_t)\Big)\Big(1+\frac{1}{10T}\Big)\right) \\
&\ \ \vdots \\
&\leq \sum_{r\in\{0,1\}^T} \frac{1}{2^T}\Big(1+\frac{1}{10T}\Big)^T \\
&\leq \Big(1+\frac{1}{10T}\Big)^T \\
&\leq e^{1/10}.
\end{aligned}
$$

□

Combining the claims above, L can be reduced to the following promise problem

- YES instances: x such that $|D'(x)| \geq |D(x)| \geq \frac{2}{3}K$;
- NO instances: x such that $|D(x)| \leq |D'(x)| \leq \frac{1}{3}K$.

This promise problem can be solved in AM using the set lower bound protocol of Goldwasser and Sipser [GS86]. Thus $\mathsf{L} \in \mathbf{AM}$.

4.3 DGS_s is Probability-Verifiable

By Lemma 3.1, for any approximation factor γ, if $\mathsf{gapSIVP}_{\gamma/\mu} \in \mathbf{SZK}$ for any constant $\mu > 1$, there exists a function g maps lattice bases to real numbers such that g is in $\mathbb{R}_\eta$-**TFAM** and $\lambda_n(\mathbf{B}) \leq g(\mathbf{B}) < \gamma(n)\lambda_n(\mathbf{B})$.

For any basis $\mathbf{B}$ and lattice point $\mathbf{v} \in \mathcal{L}(\mathbf{B})$, as $g \in \mathbb{R}_\eta - \mathbf{TFAM}$, the verifier can force the prover to provide a sufficiently accurate estimation of $g(\mathbf{B})$, denoted by $\hat{g}$. As $\hat{g} \approx g(\mathbf{B}) \geq \lambda_n(\mathbf{B})$, the verifier can ask the prover to provide a set of linearly independent vectors $\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_n)$ such that $\|\mathbf{W}\| \leq \hat{g}$. Here the length of a vector set, e.g. $\|\mathbf{W}\|$, is defined as the length of the longest vector in the set.

Given such a short independent vector set $\mathbf{W}$, there exists an efficient algorithm that samples from discrete Gaussian distribution $\mathcal{N}_{\mathcal{L}(\mathbf{B}),\hat{s}}$ such that $\hat{s} = \Theta(\sqrt{\log n}) \cdot \hat{g}$ [BLP+13, GPV08]. Moreover, the verifier can estimate the probability that $\mathbf{v}$ is sampled from $\mathcal{N}_{\mathcal{L}(\mathbf{B}),\hat{s}}$ using the set lower bound protocol.

Let $s(\mathbf{B}) = \Theta(\sqrt{\log n}) \cdot g(\mathbf{B})$, then $\hat{s}$ is a good estimation of $s(\mathbf{B})$. If the bias between $\hat{s}$ and $s(\mathbf{B})$ is sufficiently small, one could expect $\Pr[\mathbf{v} \leftarrow \mathcal{N}_{\mathcal{L}(\mathbf{B}),\hat{s}}] \approx \Pr[\mathbf{v} \leftarrow \mathcal{N}_{\mathcal{L}(\mathbf{B}),s(\mathbf{B})}]$.

Proof (Lemma 4.5). By Lemma 3.1, for sufficiently large n, $\mathsf{gapSIVP}_{\gamma(n)/\sqrt{\ln n}} \in \mathbf{SZK}$ implies the existence of a function g maps lattice bases to real numbers such that g is in $\mathbb{R}_\eta$-**TFAM** and $g(\mathbf{B}) \in [\lambda_n(\mathbf{B}), \gamma(n)/\sqrt{\ln(2n+4)/\pi} \cdot \lambda_n(\mathbf{B})]$. Here $n \geq 2$ is sufficiently large, as it implies $\frac{\gamma(n)/\sqrt{\ln(2n+4)/\pi}}{\gamma(n)/\sqrt{\ln n}} \geq 1.01$.

Define $s(\mathbf{B}) = \sqrt{\ln(2n+4)/\pi} \cdot g(\mathbf{B})$, thus for sufficiently large n

$$\lambda_n(\mathbf{B}) \leq \sqrt{\ln(2n+4)/\pi} \cdot \lambda_n(\mathbf{B}) \leq s(\mathbf{B}) < \gamma(n)\lambda_n(\mathbf{B}).$$

Given any basis $\mathbf{B}$, vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ and precision parameter m, the verifier can learn a good estimation on $g(\mathbf{B})$, denoted by $\hat{g}$. As $g(\mathbf{B}) \geq \lambda_n(\mathbf{B})$, the verifier could ask the prover to provide a set of linearly independent vectors of $\mathcal{L}(\mathbf{B})$, denoted by $\mathbf{W}$, such that $\|\mathbf{W}\| \leq \hat{g}$.

Given a set of linearly independent vectors $\mathbf{W}$ that $\|\mathbf{W}\| \leq \hat{g}$, there is an efficient algorithm which samples from discrete Gaussian $\mathcal{N}_{\mathcal{L}(\mathbf{B}),\sqrt{\ln(2n+4)/\pi}\cdot\hat{g}}$ [BLP+13]. Let $\mathcal{S}$ denote this sampling algorithm. Let $\hat{s} = \sqrt{\ln(2n+4)/\pi}\cdot\hat{g}$, then $\hat{s}$ is a good approximation of $s(\mathbf{B})$. Let r be the random tape in the sampling algorithm $\mathcal{S}$, then

$$\Pr[\mathbf{v} \leftarrow \mathcal{N}_{\mathcal{L}(\mathbf{B}),\hat{s}}] = \frac{\{r : \mathcal{S}(B', \hat{s}) \text{ outputs } \mathbf{v} \text{ when } r \text{ is the random input tape}\}}{2^{|r|}}.$$

We could use the set lower bound protocol to lower bound this probability $\Pr[\mathbf{v} \leftarrow \mathcal{N}_{\mathcal{L}(\mathbf{B}),\hat{s}}]$. Thus the promise problem

- YES instances: $(\mathbf{W}, \mathbf{v}, \hat{s}, \hat{p}, 1^m)$ such that $\mathbf{v} \in \mathcal{L}$, $\|\tilde{\mathbf{W}}\| \leq \frac{\hat{s}}{\sqrt{\ln(2n+4)/\pi}}$, $\hat{p} = \Pr[\mathbf{v} \leftarrow \mathcal{N}_{\mathcal{L}(\mathbf{B}),\hat{s}}]$
- NO instances: $(\mathbf{W}, \mathbf{v}, \hat{s}, \hat{p}, 1^m)$ such that $\hat{p} \geq (1 + \frac{1}{m}) \Pr[\mathbf{v} \leftarrow \mathcal{N}_{\mathcal{L}(\mathbf{B}),\hat{s}}]$

is in **AM**, as it can be solved by protocol ProbLowerBound.

AM protocol ProbLowerBound on input $(\mathbf{B}, \mathbf{v}, \hat{p}, 1^m)$

P: Send $\hat{g}$, an honest prover should send $\hat{g} = g(\mathbf{B})$
P,V: Convince the verifier that $|\hat{g} - g(\mathbf{B})| \leq c\delta \cdot g(\mathbf{B})$, where $\delta = \frac{1}{nm^2}$, c is a sufficiently small constant
P: Send $\mathbf{W} = (\mathbf{x}'_1, \ldots, \mathbf{x}'_n)$
V: Check if $\mathbf{W}$ is a basis of $\mathcal{L}(\mathbf{B})$ and $\|\tilde{\mathbf{W}}\| \leq \hat{g}$
P,V: Run the set lower bound protocol to convince the verifier that $\hat{p} \leq (1 + \frac{1}{2m}) \Pr[\mathbf{v} \leftarrow \mathcal{N}_{\mathcal{L}(\mathbf{B}),\hat{s}}]$, where $\hat{s} = \sqrt{\ln(2n+4)/\pi} \cdot \hat{g}$

To prove DGS_s is probability-verifiable, it is sufficient to show that ProbLowerBound is an **AM** protocol that estimates the probability $\Pr[\mathbf{v} \leftarrow \mathcal{N}_{\mathcal{L}(\mathbf{B}),\hat{s}}]$ with high accuracy. The estimation error of ProbLowerBound has two sources: (a) the inaccuracy of the set lower bound protocol, which introduce an $O(\frac{1}{m})$ multiplicative error; and (b) the inaccuracy when estimating $s(\mathbf{B})$. Let $\eta_{\mathbf{B}}(\mathbf{v})$ be the estimation error, the error term satisfies

$$\mathcal{N}_{\mathbf{B},s(\mathbf{B})}(\mathbf{v}) + \eta_{\mathbf{B}}(\mathbf{v}) \leq \left(1 + \frac{1}{2m}\right) \max_{|\hat{s}-s(\mathbf{B})| \leq \delta \cdot s(\mathbf{B})} \mathcal{N}_{\mathbf{B},\hat{s}}(\mathbf{v}) \tag{5}$$

To complete the proof, it is sufficient to show that $\sum_{\mathbf{v} \in \mathcal{L}(\mathbf{B})} \eta_{\mathbf{B}}(\mathbf{v}) = O(\frac{1}{m})$. By summing (5) over $\mathbf{v} \in \mathcal{L}(\mathbf{B})$,

$$1 + \sum_{\mathbf{v} \in \mathcal{L}(\mathbf{B})} \eta_{\mathbf{B}}(\mathbf{v}) \leq \left(1 + \frac{1}{2m}\right) \sum_{\mathbf{v} \in \mathcal{L}(\mathbf{B})} \max_{|\hat{s}-s(\mathbf{B})| \leq \delta \cdot s(\mathbf{B})} \mathcal{N}_{\mathbf{B},\hat{s}}(\mathbf{v}).$$

Thus it is sufficient to show

$$\sum_{\mathbf{v} \in \mathcal{L}(\mathbf{B})} \max_{|\hat{s}-s(\mathbf{B})| \leq \delta \cdot s(\mathbf{B})} \mathcal{N}_{\mathbf{B},\hat{s}}(\mathbf{v}) \leq 1 + O(\frac{1}{m}). \tag{6}$$

Which is proved as

$$\begin{aligned}\sum_{\mathbf{v}\in\mathcal{L}(\mathbf{B})}\max_{|\hat{s}-s(\mathbf{B})|\le\delta\cdot s(\mathbf{B})}\mathcal{N}_{\mathbf{B},\hat{s}}(\mathbf{v}) &= \sum_{\mathbf{v}\in\mathcal{L}(\mathbf{B})}\max_{|\hat{s}-s(\mathbf{B})|\le\delta\cdot s(\mathbf{B})}\frac{\rho_{\hat{s}}(\mathbf{v})}{\rho_{\hat{s}}(\mathcal{L}(\mathbf{B}))}\\ &\le \sum_{\mathbf{v}\in\mathcal{L}(\mathbf{B})}\frac{\max_{|\hat{s}-s(\mathbf{B})|\le\delta\cdot s(\mathbf{B})}\rho_{\hat{s}}(\mathbf{v})}{\min_{|\hat{s}-s(\mathbf{B})|\le\delta\cdot s(\mathbf{B})}\rho_{\hat{s}}(\mathcal{L}(\mathbf{B}))}\\ &\le \frac{\rho_{(1+\delta)s}(\mathcal{L}(\mathbf{B}))}{\rho_{(1-\delta)s}(\mathcal{L}(\mathbf{B}))} \qquad (7)\\ &\le (\frac{1+\delta}{1-\delta})^n\\ &= O(\frac{1}{mn})\end{aligned}$$

The last inequality is due to Lemma 2.1. □

Acknowledgments. I am grateful to my advisor, Vinod Vaikuntanathan, for getting me started on the topic of NP-hardness and separations. I am indebted to Adam Sealfon, Prashant Nalini Vasudevan, Srinivasan Raghuraman and Akshay Degwekar for their extensive help with the writing of this article. I would like to thank the anonymous reviewers for their careful reading and insightful comments.

References

[AGGM06] Akavia, A., Goldreich, O., Goldwasser, S., Moshkovitz, D.: On basing one-way functions on NP-hardness. In: Kleinberg, J.M. (ed.) Proceedings of the 38th Annual ACM Symposium on Theory of Computing, 21–23 May 2006, Seattle, WA, USA, pp. 701–710. ACM (2006)

[Ajt96] Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: Miller, G.L. (ed.) Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, 22–24 May 1996, Philadelphia, Pennsylvania, USA, pp. 99–108. ACM (1996)

[AR04] Aharonov, D., Regev, O.: Lattice problems in NP ∩ coNP. In: Proceedings of 45th Symposium on Foundations of Computer Science, FOCS 2004, 17–19 October 2004, Rome, Italy [DBL04], pp. 362–371 (2004)

[Ban93] Banaszczyk, W.: New bounds in some transference theorems in the geometry of numbers. Math. Ann. **296**(1), 625–635 (1993)

[BB15] Bogdanov, A., Brzuska, C.: On basing size-verifiable one-way functions on NP-hardness. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9014, pp. 1–6. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46494-6_1

[BL13] Bogdanov, A., Lee, C.H.: Limits of provable security for homomorphic encryption. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8042, pp. 111–128. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40041-4_7

[BLP+13] Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.) Symposium on Theory of Computing Conference, STOC 2013, 1–4 June 2013, Palo Alto, CA, USA, pp. 575–584. ACM (2013)

[Bra79] Brassard, G.: Relativized cryptography. In: 20th Annual Symposium on Foundations of Computer Science, 29–31 October 1979, San Juan, Puerto Rico, pp. 383–391. IEEE Computer Society (1979)

[BS99] Blömer, J., Seifert, J.P.: On the complexity of computing short linearly independent vectors and short bases in a lattice. In: Vitter, J.S., Larmore, L.L., Leighton, F.T (eds.) Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, 1–4 May 1999, Atlanta, Georgia, USA, pp. 711–720. ACM (1999)

[BT06] Bogdanov, A., Trevisan, L.: On worst-case to average-case reductions for NP problems. SIAM J. Comput. **36**(4), 1119–1159 (2006)

[DBL04] In: Proceedings of 45th Symposium on Foundations of Computer Science, FOCS 2004, 17–19 October 2004, Rome, Italy. IEEE Computer Society (2004)

[FF91] Feigenbaum, J., Fortnow, L.: On the random-self-reducibility of complete sets. In: Proceedings of the Sixth Annual Structure in Complexity Theory Conference, 30 June–3 July 1991, Chicago, Illinois, USA, pp. 124–132. IEEE Computer Society (1991)

[GG98] Goldreich, O., Goldwasser, S.: On the possibility of basing cryptography on the assumption that $P \neq NP$. IACR Cryptol. Eprint Arch. **1998**, 5 (1998)

[GG00] Goldreich, O., Goldwasser, S.: On the limits of nonapproximability of lattice problems. J. Comput. Syst. Sci. **60**(3), 540–563 (2000)

[GMR04] Guruswami, V., Micciancio, D., Regev, O.: The complexity of the covering radius problem on lattices and codes. In: 19th Annual IEEE Conference on Computational Complexity, CCC 2004, 21–24 June 2004, Amherst, MA, USA, pp. 161–173. IEEE Computer Society (2004)

[GPV08] Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Dwork, C. (ed.) Proceedings of the 40th Annual ACM Symposium on Theory of Computing, 17–20 May 2008, Victoria, British Columbia, Canada, pp. 197–206. ACM (2008)

[GS86] Goldwasser, S., Sipser, M.: Private coins versus public coins in interactive proof systems. In: Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing, pp. 59–68. ACM (1986)

[GV99] Goldreich, O., Vadhan, S.: Comparing entropies in statistical zero knowledge with applications to the structure of SZK. In: Proceedings of Fourteenth Annual IEEE Conference on Computational Complexity, pp. 54–73. IEEE (1999)

[LV16] Liu, T., Vaikuntanathan, V.: On basing private information retrieval on NP-hardness. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016. LNCS, vol. 9562, pp. 372–386. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49096-9_16

[MR04] Micciancio, D., Regev, O.: Worst-case to average-case reductions based on Gaussian measures. In: Proceedings of 45th Symposium on Foundations of Computer Science, FOCS 2004, 17–19 October 2004, Rome, Italy [DBL04], pp. 372–381 (2014)

[MV03] Micciancio, D., Vadhan, S.P.: Statistical Zero-knowledge proofs with efficient provers: lattice problems and more. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 282–298. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_17

[MX10] Mahmoody, M., Xiao, D.: On the power of randomized reductions and the checkability of SAT. In: 2010 IEEE 25th Annual Conference on Computational Complexity (CCC), pp. 64–75. IEEE (2010)

[PV08] Peikert, C., Vaikuntanathan, V.: Noninteractive statistical zero-knowledge proofs for lattice problems. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 536–553. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85174-5_30

[Reg09] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM **56**(6), 34:1–34:40 (2009)