On Basing Search SIVP on NP-Hardness

Tianren Liu
MIT*

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 got so far was 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 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 show that any language that can be reduced to solving search SIVP with approximation factor $\tilde{O}(n)$ lies in **AM** intersect **coAM**, eliminating the possibility of basing current constructions on NP-hardness.

1 Introduction

It is a long standing open question as to 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 an polynomial-time algorithm breaking the security of the primitive, one can efficiently solve SAT.

A potential line of attack is lattice cryptography. The 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_λ), which can be stated as follows: given an n-dimensional lattice, find a set of n linear independent vectors whose length[1] is at most $\gamma(n)$ (polynomial in n) times long compare to the shortest such vectors set. Since the work of Ajtai, the state of the art constructions are those of a family of collision resistant hash functions (CRHF) based on the hardness of shortest independent vectors problem with an approximation factor $\tilde{O}(n)$ [MR04]. One would hope that this approach is viable since, Khot shows SIVP_γ is NP-hard for any constant factor [Kho05]. Presumably, if one could construct cryptographic primitives based on the hardness of SIVP_γ with $\gamma(n)$ being constant, we would golden. Alternatively, if one could extend the result of Khot 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.

*liutr@mit.edu

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

However, there seem to be some sort of negative results when one considers the corresponding gap versions 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 [MX10].

Fortunately, the hardness of SIVP is not contradicted by the fact that the gap problem with same approximation factor is easy. For instance, if one considers any ideal lattice in the field $\mathbb{Z}[x]/x^{2^k}+1$, 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 not known to be easy. 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, with the gap version gapSIVP. Informally, we show that if $\mathsf{gapSIVP}_\gamma$ is not **NP**-hard, neither is $\mathsf{SIVP}_{\sqrt{n}\cdot\gamma}$.

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

Corollary (w/ [PV08])**.** *If there exists a probabilistic polynomial-time adaptive reduction from a language* L *to* $\mathsf{SIVP}_{\tilde{O}(n)}$*, then* $\mathsf{L} \in$ **AM** $\cap$ **coAM**.

As a quick corollary from combining our result with $\mathsf{gapSIVP}_{\tilde{O}(\sqrt{n})} \in$ **SZK** [PV08], any language that can be reduced to $\mathsf{SIVP}_{\tilde{O}(n)}$ lies in **AM** intersect **coAM**, thus it's not **NP**-hard unless the polynomial hierarchy collapses. This eliminates the possibility of basing current constructions on NP-hardness.

1.1 Proof Overview

The first step is to shift from a search problem to a sampling problem. We are looking for 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. A naïve verifier simulates the reduction $\mathcal{R}$ and resolves any query to SIVP_γ relying on the prover. While SIVP_γ is a search problem, and there is no unique right answer. The prover has the freedom to decide which answer is present upon each query, and this freedom allows a malicious prover to fool the naïve verifier. Bogdanov and Brzuska inherently shift to sampling problems in order to separate size-verifiable one-way functions from NP-Hardness [BB15]. The honest prover replies each query by sampling over all valid answers uniformly at random. Due to the size-verifiable property, a malicious prover has little freedom to modify the reply distribution.

The distribution discussed in this work is discrete Gaussian distribution. Discrete Gaussian over a lattice is a distribution such that the probability of vertex $\mathbf{v}$ is proportional to $e^{-\pi\|\mathbf{v}-\mathbf{c}\|^2/s^2}$, where $\mathbf{c}$ is its "centre" and parameter s is its "width". Lemma 4.1 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" $\gamma(n)/\sqrt{n}$.

- (Lemma 4.1) For $\gamma \geq \sqrt{n}$, SIVP_γ can be reduced to discrete Gaussian sampling on lattice with "width" $\sigma = \frac{\gamma}{\sqrt{n}}\lambda_n$. Therefore, if a language can be reduce to SIVP_n, then it could be reduced to discrete Gaussian sampling on lattice with "width" $s \leq \sqrt{n} \cdot \lambda_n$.

The proof of Lemma 4.1 is quite intuitive. If you can sample discrete Gaussian distributions, keep sampling multiple times from the discrete Gaussian centered at $\mathbf{0}$. With good probability, the newly sampled vertex is short and is linear 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-verifiable". Informally, a distribution family is *probability-verifiable* if for any distribution $\mathcal{D}$ in this distribution family and for any possible value v, the probability that $\Pr[v \leftarrow \mathcal{D}]$ can be computed with an arbitrarily good precision in **AM**.

- (Lemma 4.2) If a language L can be reduced to a probability-verifiable sampling problem S, then $\mathsf{L} \in \mathbf{AM} \cap \mathbf{coAM}$.

Lemma 4.2 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 $\mathcal{P}_{\mathsf{pd}}$ denotes the distribution specified by input pd. For simplicity, assume 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-verifiable.

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, denoted by r, and the how the queries to sampling problem S are answered. The *transcript* is defined as $\sigma = (r, \mathsf{pd}_1, v_1, \ldots, \mathsf{pd}_T, v_T)$ where pd_t, v_t are the t-th query/answer to S. Note that $r, v_1, \ldots, v_T$ determine the execution, pd_t is determined by $r, v_1, \ldots, v_{t-1}$ and

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

The probability $\Pr[\mathcal{R}^{\mathsf{S}}(x) \to 1]$ can be bounded by set lower bound protocol of Goldwasser and Sipser [GS86], thus $\mathsf{L} \in \mathbf{AM}$. Symmetrically, $\mathsf{L} \in \mathbf{coAM}$.

There is one last step missing between Lemma 4.1 and Lemma 4.2: *Is discrete Gaussian sampling probability-verifiable? What's the smallest factor γ such that discrete Gaussian sampling with "width" $\leq \gamma\lambda_n$ is probability-verifiable?* Lemma 4.3 answers this question, and it connects the hardness of discrete Gaussian sampling with the hardness of $\mathsf{gapSIVP}$.

- (Lemma 4.3) 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 4.3 has an easier proof if assuming a stronger condition that $\mathsf{gapSIVP}_\gamma$ is in **P**. If there is an imaginary deterministic polynomial time algorithm solving $\mathsf{gapSIVP}_\gamma$, there exists $s(\mathbf{B}) \in [\lambda_n(\mathbf{B}), \gamma\lambda_n(\mathbf{B})]$ that can be efficiently computed by binary search. As $s(\mathbf{B}) \geq \lambda_n(\mathbf{B})$, the verifier can force the prover to provide n linear independent vectors $\mathbf{w}_1, \ldots, \mathbf{w}_n$ whose length no longer than $s(\mathbf{B})$. Given the lattice basis $\mathbf{B}$ and a set of short linear independent vectors, there exists an efficient algorithm that samples from discrete Gaussian with parameter we interested in [BLP+13]. When the verifier can sample from a distribution, he can estimate the probability of each value using set lower bound protocol [GS86].

This informal proof shows that the only ability of the verifier is to compute some function $s(\mathbf{B}) \in [\lambda_n(\mathbf{B}), \gamma\lambda_n(\mathbf{B})]$. Lemma 3.1 enables the verifier to compute such a function given $\mathsf{gapSIVP}_\gamma \in \mathbf{SZK}$.

- (Corollary of Lemma 3.1) 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 **AM**.

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 linear 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 linear 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 $\lambda_n(\mathbf{B}) \ge \lambda(n) \cdot s$.

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 $\rho_{\mathbf{0},s}, \mathcal{N}_{\mathcal{L},\mathbf{0},s}$ resp.

Sampling Problems Besides computational problems and decision problems, we define *sampling problems*. The input of a sampling problem specifies a distribution, let $\mathcal{P}_{\mathsf{pd}}$ denotes 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 $\mathcal{P}_x(v)$ for any $\mathcal{P}_x$ and v. More precisely, there exists a family of error function $\{\eta_{x,m}\}$ such that for any x, m, the error function $\eta_{x,m} : \{0,1\}^* \to [0, +\infty)$ satisfies $\sum_v \eta_{x,m}(v) \leq \frac{1}{m}$, and the promise problem

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

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 complexists. 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 outputs 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 that samples from S in the sense that

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

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** This complexity class is introduced by Mahmoody and Xiao [MX10]. Informally, it's consist of real-valued functions that can be efficiently computed in **AM**. A function $\{0,1\}^* \to \mathbb{R}$ 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 $|y - f(x)| > \frac{1}{m}$.

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 input. 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 to estimate $\lambda_n(\mathbf{B})$ 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. The easyness of $\mathsf{gapSIVP}_\gamma$ (the gap problem of function λ_n) 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 $f(x) \leq g(x) \leq \lambda(x) \cdot f(x)$ 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 $\mathbb{R}$-**TFAM**.*

Lemma 3.1 can be combined with previous results about gapSIVP. Guruswami, et al. show that $\mathsf{gapSIVP}_{\sqrt{n/\log n}} \in$ **coAM** [GMR04]. Peikert and Vaikuntanathan show that $\mathsf{gapSIVP}_\gamma \in$ **NISZK** $\subseteq$ **SZK** for any $\gamma = \omega(\sqrt{n \log n})$ [PV08]. Thus there exists an approximation of λ_n within a factor $\tilde{O}(\sqrt{n})$ that can be computed in **AM**.

Corollary (w/ [PV08])**.** *For any $\gamma(n) = \omega(\sqrt{n \log n})$, there exists a function g maps lattice bases to real numbers such that $g \in \mathbb{R}$-**TFAM** and $\lambda_n(\mathbf{B}) \leq g(\mathbf{B}) < \gamma(n) \cdot \lambda_n(\mathbf{B})$.*

Proof of Lemma 3.1. Entropy Difference (ED) is a complete problem for **SZK**, so $\mathsf{gap}f_\gamma \in$ **SZK** implies the existance 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

$$\operatorname{clamp}(y) := \begin{cases} 0, & \text{if } y \le 0; \\ y, & \text{if } y \in (0,1); \\ 1, & \text{if } y \ge 1. \end{cases}$$

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

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

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

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

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

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

In order to complete the proof, we show that g is in $\mathbb{R}$-**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.

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 $\hat{d}_i - \varepsilon < H(\mathcal{C}_{x,\mu^i}) - H(\mathcal{D}_{x,\mu^i}) < \hat{d}_i + \varepsilon$

On any input x, define $d_i = H(\mathcal{C}_{x,\mu^i}) - H(\mathcal{D}_{x,\mu^i})$. And 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 $f(x) < \mu^i < \mu\gamma(x)f(x)$, $|\operatorname{clamp}(\hat{d}_i) - \operatorname{clamp}(d_i)| \le |\hat{d}_i - d_i| < \varepsilon$.

Thus

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

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's 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 $d_j \le -1$.

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

- $\log_\mu \hat{g} = i_L + \sum_{i=i_L}^{i_H} \mathrm{clamp}(\hat{d}_i)$
- $i_H = i_L + 2\lceil \log_\mu \gamma(x) \rceil$
- $\hat{d}_{i_L + \lceil \log_\mu \gamma(x) \rceil} > 0$
- $\hat{d}_{i_L + \lceil \log_\mu \gamma(x) \rceil + 1} < 1$

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

Lemma 4.1. *Let $s(\cdot)$ be a function mapping lattice bases to real numbers, such that $\forall \mathbf{B}, 2\lambda_n(\mathbf{B}) \leq s(\mathbf{B}) \leq \frac{\gamma}{\sqrt{n}}\lambda_n(\mathbf{B})$. Then there exists a probabilistic Turing reduction from SIVP_γ to DGS_s.*

Lemma 4.2. *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 4.3. *For any factor γ, if $\mathsf{gapSIVP}_{\gamma(n)/\sqrt{\pi \log(2n+4)}} \in \mathbf{SZK}$, then there exists a function $s(\cdot)$ mapping lattice bases to real numbers, such that $\forall \mathbf{B}, s(\mathbf{B}) \in [2\lambda_n(\mathbf{B}), \gamma(n) \cdot \lambda_n(\mathbf{B})]$ and DGS_s is probability-verifiable.*

4.1 From Search SIVP to Discrete Gaussian Sampling

This section proves Lemma 4.1. The reduction from SIVP_γ to discrete Gaussian sampling is straightforward: If we can sample from discrete Gaussian distribution of width $s \in [2 \cdot \lambda_n, \frac{\gamma}{\sqrt{n}}\lambda_n]$, keep sampling from it until n short, linear independent vectors are sampled.

When sampling from discrete Gaussian $\mathcal{N}_{\mathcal{L}(\mathbf{B}),s(\mathbf{B})}$, there are two bad events that might occurs:

- The sampled vector is too long, its Euclidean norm is larger than $\gamma\lambda_n(\mathbf{B})$.
- The sampled vector is not linearly independent from previous ones, it lies in the subspace spanned by previously chosen vectors.

In order to prove Lemma 4.1, it's sufficient to show that there is a constant probability that none of these bad events occurs.

Lemma 4.4 (Lemma 1.5 in [Ban93])**.** *For any $c > 1/\sqrt{2\pi}$, n-dimensional lattice $\mathcal{L}$*

$$\rho_s(\mathcal{L} \setminus cs\sqrt{n}\mathcal{B}) < C^n \cdot \rho_s(\mathcal{L}) \tag{1}$$

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

Lemma 4.4 bounds the probability that an overlong vector is sampled from a discrete Gaussian distribution. Let the constant c in formula (1) equals 1,

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

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

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

For any proper linear subspace $\mathcal{V}\subsetneq\mathbb{R}^n$, we would bound the probability $\Pr_{\mathbf{v}\leftarrow\mathcal{N}_{\mathcal{L}(\mathbf{B}),s(\mathbf{B})}}[\mathbf{v}\in\mathcal{V}]$. By the definition of successive minimum, there exists $\mathbf{u}\in\mathcal{L}\setminus\mathcal{V}$ such that $\|\mathbf{u}\|\leq\lambda_n(\mathbf{B})$. Let $\mathcal{L}'$ denotes $\mathcal{L}\cap\mathcal{V}$. As $\mathcal{L}$ is close under addition, $\mathcal{L}'+\mathbf{u}, \mathcal{L}'-\mathbf{u}$ are subsets of $\mathcal{L}$. Moreover, as $\mathcal{V}$ is close under addition and $\mathbf{u}\notin\mathcal{V}$, $\mathcal{L}'+\mathbf{u}, \mathcal{L}', \mathcal{L}'-\mathbf{u}$ are disjointed.

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

For any vectors $\mathbf{u},\mathbf{v}$ such that $\|\mathbf{u}\|\leq\lambda_n(\mathbf{B})\leq s/2$,

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

Thus

$$\Pr_{\mathbf{v}\leftarrow\mathcal{N}_{\mathcal{L}(\mathbf{B}),s(\mathbf{B})}}[\mathbf{v}\in\mathcal{V}] \leq \frac{\sum_{\mathbf{v}\in\mathcal{L}'}\rho_s(\mathbf{v})}{\sum_{\mathbf{v}\in\mathcal{L}'}(1+2e^{-\pi/2^2})\rho_s(\mathbf{v})} = \frac{1}{1+2e^{-\pi/2^2}} < 0.6.$$

By union bound, the probability of bad event is at most 0.8. By Chernoff bound, if sample m times from the discrete Gaussian distribution (for $m\geq 10n$), the probability that the sampled vectors contains n linear independent vectors of length at most $\gamma\lambda_n(\mathbf{B})$ is at least $1-e^{-\frac{m}{10}}$.

4.2 Probability-Verifiable Sampling Problem and NP-hardness

This section proves Lemma 4.2, which is a generalization of [BB15], the proof technique 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 estimate this sum.

Proof of Lemma 4.2 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

$$\begin{aligned} x \in \mathsf{L}_Y &\implies \Pr[\mathcal{M}^{\mathcal{S}}(x) \to 1] \geq \frac{8}{9}, \\ x \in \mathsf{L}_N &\implies \Pr[\mathcal{M}^{\mathcal{S}}(x) \to 1] \leq \frac{1}{9}. \end{aligned} \tag{2}$$

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 $P_x(\sigma)$ denotes 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 (2), 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 $\sum_v \eta_{\mathsf{pd},m}(v) \le 1$, 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 $\hat{p} \ge \mathcal{P}_{\mathsf{pd}}(v) + \frac{1}{m}\eta_{\mathsf{pd},m}(v)$

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

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

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 $\mathsf{ProbLowerBound}(\mathsf{pd}_t, v_t, 1^{10T})$ for all $1 \le t \le T$, 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 $1 \le k \le K \cdot \frac{1}{2^T} \prod_{i=1}^{q} \hat{p}_i$; Reject if not

---

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

For malicious prover, he should send $\hat{p}_t$ such that $\hat{p}_t \le \mathcal{P}_{\mathsf{pd}_t}(v_t) + \frac{1}{10T}\eta_{\mathsf{pd}_t,10T}(v_t)$, otherwise he'll be caught in ProbLowerBound protocol with overwhelming probability. Thus he could not let verifier accept (x, σ, k) 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,

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

□

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

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

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 $\mathbb{R}$-$\mathbf{TFAM}$ and $\lambda_n(\mathbf{B})\le g(\mathbf{B})<\gamma(n)\lambda_n(\mathbf{B})$.

For any base $\mathbf{B}$ and lattice point $\mathbf{v}\in\mathcal{L}(\mathbf{B})$. As $g\in\mathbb{R}$-$\mathbf{TFAM}$, the verifier can force the prover to provide a sufficiently accurate estimation of $g(\mathbf{B})$, denoted by $\hat{g}$. As $\hat{g}\approx g(\mathbf{B})\ge\lambda_n(\mathbf{B})$, the verifier can ask the prover to provide a set of linear independent vectors $\mathbf{W}=(\mathbf{w}_1,\ldots,\mathbf{w}_n)$ such that $\|\mathbf{W}\|\le\hat{g}$. 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 $\hat{s}=\varphi(n)\cdot\hat{g}$ and $\varphi(n)=\Theta(\sqrt{n\log n})$ [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 set lower bound protocol.

Let $s(\mathbf{B})=\varphi(n)\cdot g(\mathbf{B})$, 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 of Lemma 4.3. By Lemma 3.1, $\mathsf{gapSIVP}_{\gamma(n)/\sqrt{\pi \log(2n+4)}} \in \mathbf{SZK}$ implies the existence of a function g maps lattice bases to real numbers such that g is in $\mathbb{R}$-**TFAM** and $g(\mathbf{B}) \in [3 \cdot \lambda_n(\mathbf{B}), \gamma(n)/\sqrt{\log(2n+4)/\pi} \cdot \lambda_n(\mathbf{B})]$. Define $s(\mathbf{B}) = \sqrt{\ln(2n+4)/\pi} \cdot g(\mathbf{B})$, thus

$$2\lambda_n(\mathbf{B}) \leq 3 \cdot \sqrt{\ln(2n+4)/\pi} \cdot \lambda_n(\mathbf{B}) \leq s(\mathbf{B}) < \gamma(n)\lambda_n(\mathbf{B}).$$

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 linear independent vectors of $\mathcal{L}(\mathbf{B})$, denoted by $\mathbf{W}$, such that $\|\mathbf{W}\| \leq \hat{g}$.

Given a set of linear independent vectors $\mathbf{W}$ that $\|\tilde{\mathbf{W}}\| \leq \hat{g}$, 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 set lower bound protocol to lower bound the 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 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's sufficient to show that ProbLowerBound is an **AM** protocol that estimate 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 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

$$\mathcal{N}_{\mathbf{B},s(\mathbf{B})}(\mathbf{v}) + \eta_{\mathbf{B}}(\mathbf{v}) \leq \left(1 + \frac{1}{2m}\right) \max_{|\hat{s}-s(\mathbf{B})| \leq \delta \cdot s(\mathbf{B})} \mathcal{N}_{\mathbf{B},\hat{s}}(\mathbf{v}) \tag{3}$$

To complete the proof, it's sufficient to show that $\sum_{\mathbf{v} \in \mathcal{L}(\mathbf{B})} \eta_{\mathbf{B}}(\mathbf{v}) = O(\frac{1}{m})$. By summing (3) 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's sufficient to show

$$\sum_{\mathbf{v}\in\mathcal{L}(\mathbf{B})} \max_{|\hat{s}-s(\mathbf{B})|\le\delta\cdot s(\mathbf{B})} \mathcal{N}_{\mathbf{B},\hat{s}}(\mathbf{v}) \le 1 + O(\frac{1}{m}). \tag{4}$$

$$\begin{aligned}
\sum_{\mathbf{v}\in\mathcal{L}(\mathbf{B})} \max_{|\hat{s}-s(\mathbf{B})|\le\delta\cdot s(\mathbf{B})} \mathcal{N}_{\mathbf{B},\hat{s}}(\mathbf{v}) &= \sum_{\mathbf{v}\in\mathcal{L}(\mathbf{B})} \max_{|\hat{s}-s(\mathbf{B})|\le\delta\cdot s(\mathbf{B})} \frac{\rho_{\hat{s}}(\mathbf{v})}{\rho_{\hat{s}}(\mathcal{L}(\mathbf{B}))} \\
&\le \sum_{\mathbf{v}\in\mathcal{L}(\mathbf{B})} \frac{\max_{|\hat{s}-s(\mathbf{B})|\le\delta\cdot s(\mathbf{B})} \rho_{\hat{s}}(\mathbf{v})}{\min_{|\hat{s}-s(\mathbf{B})|\le\delta\cdot s(\mathbf{B})} \rho_{\hat{s}}(\mathcal{L}(\mathbf{B}))} \\
&\le \frac{\rho_{(1+\delta)s}(\mathcal{L}(\mathbf{B}))}{\rho_{(1-\delta)s}(\mathcal{L}(\mathbf{B}))}
\end{aligned} \tag{5}$$

For any short vector $\mathbf{v}$ that $\|\mathbf{v}\| \ll s/\sqrt{\delta}$, the relative difference between $\rho_{(1+\delta)s}(\mathbf{v})$ and $\rho_{(1-\delta)s}(\mathbf{v})$ is small.

$$\frac{\rho_{(1+\delta)s}(\mathbf{v})}{\rho_{(1-\delta)s}(\mathbf{v})} = \exp\left(\frac{\pi\|\mathbf{v}\|_2^2}{s^2}\left((1-\delta)^{-2} - (1+\delta)^{-2}\right)\right) = 1 + O\left(\delta\cdot\frac{\|\mathbf{v}\|_2^2}{s^2}\right)$$

Let radius $r = s\cdot\sqrt{n}\cdot\log m$, then

$$\frac{\rho_{(1+\delta)s}(\mathcal{L}(\mathbf{B})\cap r\mathcal{B})}{\rho_{(1-\delta)s}(\mathcal{L}(\mathbf{B})\cap r\mathcal{B})} = 1 + O\left(\delta\cdot\frac{r^2}{s^2}\right) = 1 + O((\log m)^2/m^2) = 1 + o(1/m) \tag{6}$$

For long vectors in lattice $\mathcal{L}(\mathbf{B})$, the sum of their probability in distribution $\mathcal{N}_{\mathcal{L}(\mathbf{B}),(1+\delta)s}$ is small. In particular, by lemma 4.4

$$\begin{aligned}
\frac{\rho_{(1+\delta)s}(\mathcal{L}(\mathbf{B}))}{\rho_{(1+\delta)s}(\mathcal{L}(\mathbf{B})\cap r\mathcal{B})} &= 1 + \frac{\rho_{(1+\delta)s}(\mathcal{L}(\mathbf{B})\setminus r\mathcal{B})}{\rho_{(1+\delta)s}(\mathcal{L}(\mathbf{B})\cap r\mathcal{B})} \\
&\le 1 + O(\log m\cdot e^{-\pi(\log m)^2}) \\
&= 1 + o(1/m).
\end{aligned} \tag{7}$$

Inequality (4) is proved by combining (5)(6)(7),

$$\begin{aligned}
\sum_{\mathbf{v}\in\mathcal{L}(\mathbf{B})} \max_{|\hat{s}-s(\mathbf{B})|\le\delta\cdot s(\mathbf{B})} \mathcal{N}_{\mathbf{B},\hat{s}}(\mathbf{v}) &\le \frac{\rho_{(1+\delta)s}(\mathcal{L}(\mathbf{B}))}{\rho_{(1-\delta)s}(\mathcal{L}(\mathbf{B}))} \\
&\le \left(1 + o(\frac{1}{m})\right)\cdot\frac{\rho_{(1+\delta)s}(\mathcal{L}(\mathbf{B})\cap r\mathcal{B})}{\rho_{(1-\delta)s}(\mathcal{L}(\mathbf{B})\cap r\mathcal{B})} \\
&\le \left(1 + o(\frac{1}{m})\right)\left(1 + o(\frac{1}{m})\right)
\end{aligned}$$

□

References

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

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

[BB15] Andrej Bogdanov and Christina Brzuska. On basing size-verifiable one-way functions on np-hardness. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I*, volume 9014 of *Lecture Notes in Computer Science*, pages 1–6. Springer, 2015.

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 575–584. ACM, 2013.

[GMR04] Venkatesan Guruswami, Daniele Micciancio, and Oded Regev. 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*, pages 161–173. IEEE Computer Society, 2004.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 197–206. ACM, 2008.

[GS86] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 59–68. ACM, 1986.

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

[Kho05] Subhash Khot. Hardness of approximating the shortest vector problem in lattices. *J. ACM*, 52(5):789–808, 2005.

[MR04] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, pages 372–381. IEEE Computer Society, 2004.

[MX10] Mohammad Mahmoody and David Xiao. On the power of randomized reductions and the checkability of sat. In *Computational Complexity (CCC), 2010 IEEE 25th Annual Conference on*, pages 64–75. IEEE, 2010.

[PV08] Chris Peikert and Vinod Vaikuntanathan. Noninteractive statistical zero-knowledge proofs for lattice problems. In David Wagner, editor, *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, volume 5157 of *Lecture Notes in Computer Science*, pages 536–553. Springer, 2008.