Shortest Lattice Vectors in the Presence of Gaps *

Mingjie Liu[1], Xiaoyun Wang[1,2], Guangwu Xu[1,3] and Xuexin Zheng[2]

[1] Institute for Advanced Study, Tsinghua University, Beijing 100084, China
liu-mj07@mails.tsinghua.edu.cn,xiaoyunwang@mail.tsinghua.edu.cn
[2] Key Laboratory of Cryptologic Technology and Information Security,
Ministry of Education, Shandong University, Jinan 250100, China
[3] Department of Electrical Engineering and Computer Science, University of Wisconsin-Milwaukee,
Milwaukee, WI 53201, USA
gxu4uwm@uwm.edu,zhxuexin@mail.sdu.edu.cn

Abstract. Given a lattice $\mathcal{L}$ with the i-th successive minimum λ_i, its i-th gap $\frac{\lambda_i}{\lambda_1}$ often provides useful information for analyzing the security of cryptographic schemes related to $\mathcal{L}$. The paper concerns short vectors for lattices with gaps. In the first part, a λ_2-gap estimation of LWE lattices with cryptographic significance is given. Under some conditions, a better reduction from $BDD_{\gamma'}$ to $uSVP_\gamma$ is obtained in the presence of larger λ_2-gap. The second part of the paper shows that gaps among the successive minima lead to a more efficient SVP search algorithm. As far as we know, it is the first SVP algorithm in allusion to lattices with gaps.

Key words: lattice, successive minima, approximate SVP, gap, LWE problem,

1 Introduction

The security of lattice-based cryptographic schemes relies on the hard problems in computational lattice theory such as SVP (shortest vector problem) and CVP (closest vector problem). Cryptanalysis of lattice-based schemes focuses on the reductions between different hard problems and the fast algorithms for SVP, CVP, and their approximate variants.

Successive minima in a lattice, as it will be explicitly defined later, is the sequence $\{\lambda_i\}$ of minimal lengths modulo dimension. In this paper, we emphasize that λ_i-gap (defined as λ_i/λ_1) provides additional information in cryptanalysis. The first provable lattice cryptosystem proposed by Ajtai[1] is based on the hardness of solving the worst-case uSVP$_\gamma$ problem (SVP for lattice with $\lambda_2/\lambda_1 > \gamma$ for $\gamma = n^c$). Lyubashevsky and Micciancio[18] introduced a reduction from $GapSVP_{\gamma\sqrt{n/\log n}}$(a decision variant of approximate SVP) to $uSVP_\gamma$ which strengthens cryptographers' confidence in the hardness of uSVP. They also proved that the bounded distance decoding problem(BDD, a special case of closest vector problem) can be reduced to uSVP problem by the widely used embedding technique of Kannan [15].

It is obvious that the hardness of uSVP$_\gamma$ depends on the size of γ. Some cryptographic lattices are found to have serious security problems because of the large gap between λ_2 and λ_1(see [8]). For examples, the knapsack lattice with evident λ_2-gap results in the low density attack on the general knapsack cryptosystems [19,5]; and the large λ_2-gap in embedding lattice of GGH [27] makes the shortest vector easy to search. Even for the high density knapsack lattice with gap close to 1, there is a broadcast attack to enlarge the gap [32]. For the public-key cryptosystem NTRU, Coppersmith and Shamir [6] constructed a cryptographic lattice with dimension $2N$ to analyze its security. A

* Supported by the National "973" Program of China (Grant No.2007CB807902) and National Natural Science Foundation of China (Grant No.60931160442).

heuristic analysis reported in [13] indicates that NTRU lattice has λ_{N+1}-gap. So it is obvious that the estimation of the λ_i-gap and fast searching algorithm for SVP of lattices with gaps are of great importance in lattice cryptanalysis.

For approximate SVP, the LLL basis reduction algorithm [17] achieves an exponential approximate factor in polynomial time, where the output vector $\mathbf{b}_1$ satisfies $\frac{\|\mathbf{b}_1\|}{\lambda_1} \leq \left((1+\varepsilon)\sqrt{\frac{4}{3}}\right)^{\frac{n-1}{2}}$. Schnorr [35] generalized LLL algorithm to block size reduction. Assuming the block size is k, then there is a balance between approximate factor $O((6k^2)^{\frac{n}{k}})$ and the running time which is determined by calling $ploy(n)$ times k-dimensional exact SVP algorithm. To the best of our knowledge, the fastest method for approximate SVP is the one proposed by Gama and Nguyen [9]. Given a basis of a n-dimensional lattice, a block size k dividing n, a reduction factor $\varepsilon > 0$, and a k-dimensional SVP oracle, the algorithm in [9] outputs a basis with the first vector $\mathbf{b}_1$ satisfying $\frac{\|\mathbf{b}_1\|}{\lambda_1} \leq ((1+\varepsilon)\gamma_k)^{\frac{n-k}{k-1}}$, where γ_n is Hermite's constant [7,22]. The number of calls to the SVP-subroutine is $poly(n)$. When $k = \alpha n$, the approximate factor decreases to $poly(n)$ with $2^{O(n)}$ operations. In practice, basis reduction algorithms behave better than their proved worst-case theoretical bounds. In [8], Gama and Nguyen gave a significative assessment on the actual behaviour of lattice reduction algorithms, based on extensive experiments.

For exact SVP, there are many deterministic enumeration algorithms [10,14,30,36], with computational time ranging from $2^{O(n^2)}$ down to $2^{O(n \log n)}$ and with polynomial space. The random sieve for SVP was first proposed by Ajtai et.al in [2]. This algorithm, known as AKS sieve, gives a significant improvement in cutting the time complexity to $2^{O(n)}$ with $2^{O(n)}$ space. The algorithm invokes (polynomially many times) a sieve to compress the set of lattice points from big ball of radius R to that of γR ($\gamma < 1$) without shrinking the size of the set too much. Micciancio and Voulgaris [25] introduced another type of random sieve algorithm named ListSieve, which solves SVP in time $2^{3.199n+o(n)}$ and space $2^{1.325n+o(n)}$, and soon the time complexity was improved to $2^{2.465n}$ by Pujol and Stehlé [31] using the birthday attack. Also, under some random assumptions, there are heuristic versions of AKS sieve algorithm[28,38] and ListSieve[25] which are more efficient in practice. The best exact deterministic algorithm for most lattice problems, including SVP, SMP(Successive Minima Problem), CVP was recently proposed by Micciancio and Voulgaris [26] based on Voronoi cell computation, whose running time is $\widetilde{O}(2^{2n+o(n)})$ [4].

This paper concerns short vectors for lattices with gaps. Firstly we estimate the tight low bound of the λ_2-gap for an embedding lattice from LWE problem. For one set of recommended parameters, the λ_2-gap of this lattice is larger than $7.3 \log^2 m$, where m is the lattice dimension. Furthermore, combining with the evaluation on performance of lattice reduction [8], we estimate the range of the crucial parameter that can be attacked by finding the shortest vector in the embedding lattice. This analysis provides a useful security assessment of the schemes based on LWE. Also, under some conditions, a better reduction from BDD_{γ_1} to $uSVP_\gamma$ with larger λ_2-gap for some value of γ_1 is given.

Secondly, we show that gaps among successive minima lead to a more efficient SVP search algorithm. Our purpose is to convert the n-dimensional SVP to the SVP of a sublattice of reduced-dimension, and then invoke the best deterministic SVP algorithm described in [26] to find the solution. The complexity is determined by the location of the gap and its size. For example, when $\lambda_{0.4n+1} > c(n)\lambda_1$, and $\lim_{n\to\infty} c(n) = \infty$, the time complexity decreases to $2^{0.802n}$. This implies

[4] One writes $f = \tilde{O}(g)$ if $f(n) \leq \left(\log^c g(n)\right) \cdot g(n)$ for some constant c.

that the occurrence of gap is a big security problem not only between λ_2 and λ_1, but also between other successive minima. Although, it is believed there is no gap in random lattice, cryptographic lattices tend to possess this special property. As far as we know, it is the first SVP algorithm in allusion to lattices with gaps.

This paper is organized as follows: Section 2 is the preliminaries where some notations and useful facts are included. In Section 3, we exhibit the λ_2-gaps of cryptographic lattices obtained from LWE problem. Furthermore, for some parameters, an improved reduction from BDD to uSVP is presented. The SVP algorithm for lattices with gaps among successive minima is described in Section 4. Conclusions are given in Section 5.

2 Notations and Background

Let $B = \{\mathbf{b_1}, \ldots, \mathbf{b_n}\} \subseteq \mathbb{R}^m$ consist of n linearly independent vectors. The n-dimensional lattice generated by the basis B is defined as,

$$\mathcal{L}(B) = \left\{ \sum_{i=1}^{n} x_i \mathbf{b_i} : x_i \in \mathbb{Z} \right\}.$$

The fundamental parallelepiped $\mathcal{P}(B)$ is defined to be $\{\Sigma_i x_i \mathbf{b}_i : 0 \leq x_i < 1\}$ and $\mathcal{P}_{1/2}(B) = \{\Sigma_i x_i \mathbf{b}_i : -\frac{1}{2} \leq x_i < \frac{1}{2}\}$. The volume of $\mathcal{P}(B)$ is called determinant of the lattice.

Some basic notations used in this paper include:

- $\|\mathbf{x}\|$ is the Euclidean norm of a vector $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$, i.e, $\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$.
- $\|\mathbf{x}\|_\infty$ is the l_∞ norm of a vector $\mathbf{x} \in \mathbb{R}^n$, i.e., $\|\mathbf{x}\|_\infty = \max |x_i|$.
- $B_n(r)$ denotes the n-ball centered at the origin with radius r.
- $N(n, R^2)$ is the number of integer points in $B_n(R)$, i.e.,

$$N(n, R^2) = \left| \left\{ z \in \mathbb{Z}^n : \sum_{i=1}^{n} z_i^2 \leq R^2 \right\} \right|.$$

It is known that when $R > \sqrt{n/2}$, $N(n, R^2) \approx (\frac{2\pi e}{n})^{\frac{n}{2}} R^n$ [23].

Most of the worst-case to average-case reduction are done on q-ary lattices $\mathcal{L}$ with $q\mathbb{Z}^n \subseteq \mathcal{L} \subseteq \mathbb{Z}^n$. Given a matrix $A \in \mathbb{Z}_q^{m \times n}$, for some integers q, m, n, the following two types of m-dimensional q-ary lattices will be used in our discussion.

$$\Lambda_q(A^T) = \{\mathbf{y} \in \mathbb{Z}^m : \mathbf{y} = A\mathbf{s} \mod q \text{ for } \mathbf{s} \in \mathbb{Z}_q^n\}$$

$$\Lambda_q^\perp(A^T) = \{\mathbf{y} \in \mathbb{Z}^m : \mathbf{A^T y} = \mathbf{0} \mod q\}.$$

From the definition, these lattices are dual to each other up to a factor q. More precisely $\Lambda_q(A^T) = q\Lambda_q^\perp(A^T)^*$,$\Lambda_q^\perp(A^T) = q\Lambda_q(A^T)^*$. It follows that for any $\mathbf{v} \in \Lambda_q(A^T), \mathbf{w} \in \Lambda_q^\perp(A^T), \langle \mathbf{v}, \mathbf{w} \rangle \equiv 0 \mod q$. Determinants of these two types of lattices are $det(\Lambda_q(A^T)) = q^{m-n}$ and $det(\Lambda_q^\perp(A^T)) = q^n$ respectively.

Next, we list some computational complexity problems and some results in lattice theory that are of relevance to our discussion.

- **Shortest Vector Problem (SVP)**: Given a basis of a lattice $\mathcal{L}$, find a shortest nonzero vector in $\mathcal{L}$.
- **γ-Approximate Shortest Vector Problem ($\mathbf{SVP}_\gamma$)**: Given a basis of a lattice $\mathcal{L}$, find a nonzero lattice vector $\mathbf{v}$ such that $\|\mathbf{v}\| \leq \gamma\|\mathbf{u}\|$, for any $\mathbf{u} \in \mathcal{L}$.
- **Successive Minima Problem(SMP)**: Given a basis of a lattice $\mathcal{L}$, find lattice vector $\mathbf{s}_i$ so that $\lambda_i(\mathcal{L}) = \|\mathbf{s}_i\|$, where $\lambda_i(\mathcal{L}) = \inf\{r | \dim(span(\mathcal{L} \cap B_n(r))) \geq i\}$, $i = 1, \ldots n$.
- **γ-Unique Shortest Vector Problem($uSVP_\gamma$)**: Given a basis of a lattice $\mathcal{L}$ such that $\lambda_2(\mathcal{L}) > \gamma \cdot \lambda_1(\mathcal{L})$, find the shortest nonzero lattice vector.
- **Closest Vector Problem (CVP)**: Given a basis of a lattice $\mathcal{L}$ and a target vector $\mathbf{t} \in \mathbb{R}^m$, find the lattice vector closest to $\mathbf{t}$.
- **γ-Bounded Distance Decoding (BDD)**: Given a basis of a lattice $\mathcal{L}$ and a target vector $\mathbf{t}$ such that $dist(\mathbf{t}, \mathcal{L}) < \gamma\lambda_1(\mathcal{L})$, find the lattice vector closest to $\mathbf{t}$.

Given a basis of an n-dimensional lattice, the LLL algorithm [17] produces another basis of the lattice which consists of shorter vectors and is referred to as the LLL-reduced basis. Moreover,

Lemma 1 *[17] Let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ be an LLL-reduced basis, then for any set of linear independent lattice vectors $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_t\}$, we have*

$$\|\mathbf{b}_j\| \leq 2^{\frac{n-1}{2}} \max\{\|\mathbf{w}_1\|, \ldots, \|\mathbf{w}_t\|\}, (1 \leq j \leq t).$$

In particular, when $j = t$, it is equivalent to $\|\mathbf{b}_t\| \leq 2^{\frac{n-1}{2}}\lambda_t$.

According to the LLL-reduced basis properties above, we can determine a μ_i such that $\lambda_i \leq \mu_i \leq (1 + \frac{1}{n})\lambda_i$ by trying polynomial number of values for μ_i. This approximation is essential in the reduction between BDD to uSVP in section 3.2 and our approximate SVP algorithm in section 4.1. When $i = 1$, we simply write μ for μ_1.

Predicting lattice reduction algorithms: In practice, almost all basis reduction algorithms perform better than the theoretical upper bound. In [8], based on abundant experiment, Gama and Nguyen introduced Hermit factor δ as a parameter to estimate the output of the algorithm. The basis reduction algorithm can output a short vector with length about $\delta^m(det)^{\frac{1}{m}}$ on m-dimensional lattice. δ varies in different algorithms. It is 1.0219 for LLL and 1.0128 for BKZ-20. [8] also argues that the algorithm can output the shortest vector in the case of $\lambda_2/\lambda_1 > \tilde{\delta}^m$, for some $\tilde{\delta}$ being slightly smaller than δ. In [24] and [20], the authors applied this estimation to evaluate the efficiency of their attacks on LWE. After discussing the λ_2-gap in LWE embedding lattice, we will also use it to analyze the efficiency of solving LWE by embedding method.

Computing a basis for the intersection of a lattice and a subspace should be easy, as indicated by the next lemma.

Lemma 2 *[21] There is a polynomial time algorithm, that given inputs $\mathcal{L}$ with basis $\mathbf{B} \in \mathbb{Q}^{m \times n}$ and a linear subspace $S \subset \mathbb{R}^n$, outputs a basis $\tilde{\mathbf{B}}$ for the sublattice $S \cap \mathcal{L}(\mathbf{B})$.*

3 λ_2-Gap for Some Embedding Lattices

It is well known that some popular knapsack cryptosystems are broken by lattice reduction algorithms, and some instances of lattice cryptosystems such as GGH challenges [27] and PJH [12] are also successfully attacked. The main reason is that the λ_2-gaps of lattices used to analyze these

cryptosystems are large enough to leak substantial information. In this section, we shall discuss the λ_2-gaps of the lattices used to analyze the LWE-based cryptosystem. Furthermore, we present a reduction from BDD_{γ_1} to $uSVP_\gamma$ with larger λ_2-gap for some value of γ_1.

3.1 λ_2-Gap for LWE-based Lattice

The computational infeasibility of learning with error problem (LWE) [34] guarantees the security of several cryptographic schemes[11,20,29,34]. The input to this problem is a pair $(A, \mathbf{v} = A\mathbf{s} + \mathbf{e})$ where $A \in \mathbb{Z}_q^{m\times n}$ and $\mathbf{s} \in \mathbb{Z}_q^n$ are chosen uniformly, $\mathbf{e} \in \mathbb{Z}_q^m$ is chosen according to some distribution χ. The decision version of LWE is to distinguish $\mathbf{v}$ from a vector of uniform distribution in $\mathbb{Z}_q^m$, and the search version LWE is to recover $\mathbf{s}$. LWE problem can be seen as a BDD instance in q-ary lattice $\Lambda_q(A^T) = \{\mathbf{y} \in \mathbb{Z}^m : \mathbf{y} = A\mathbf{s} \mod q \text{ for } \mathbf{s} \in \mathbb{Z}_q^n\}$. In general, the error vector $\mathbf{e}$ obeys the discrete Guassian distribution on $D_{\mathcal{L},s}$ which is defined as follows: for any real $s > 0$, n-dimensional lattice $\mathcal{L}$, $\forall \mathbf{x} \in \mathcal{L}$, $D_{\mathcal{L},s}(\mathbf{x}) = \frac{\rho_{\mathbf{s}}(\mathbf{x})}{\rho_{\mathbf{s}}(\mathcal{L})}$ where $\rho_{\mathbf{s}}(\mathbf{x}) = e^{-\pi\|\mathbf{x}/s\|^2}$ and $\rho_{\mathbf{s}}(\mathcal{L}) = \sum_{x\in\mathcal{L}} \rho_{\mathbf{s}}(\mathbf{x})$.

The hardness of LWE was studied in [34] and it is proved that for $\chi = D_{\mathbb{Z}_q^m,\alpha q}$, $\alpha q \geq 2\sqrt{n}$, the search-LWE is at least as hard as quantumly approximating a worst-case (the decision variant of) SVP or SIVP of n-dimensional lattice to within $\tilde{O}(n/\alpha)$. In [29], Peikert presented a classical reduction for (the decision variant of) SVP at the cost of increasing q. Recently, cryptanalysts proposed two methods to solve LWE problem. One of them is to find a short vector in its dual lattice to distinguish $\mathbf{v}$ from uniform distribution [24], the other is to use a variant of NearestPlane algorithm [20] to search-LWE.

Let us briefly describe the embedding technique that is used to reduce CVP to SVP in [15]. Given a lattice $\mathcal{L}$ with basis $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_m]$, and a target vector $\mathbf{t} \in span(B)$, the embedding method is to construct a new lattice $\mathcal{L}'$ with basis $\mathbf{B}' = [\mathbf{b}'_1, \mathbf{b}'_2, \cdots, \mathbf{b}'_{m+1}]$ as follows:

$$\begin{aligned}
\mathbf{b}'_{\mathbf{1}} &= (b_{11}, b_{12}, \ldots, b_{1m}, 0)\\
&\vdots\\
\mathbf{b}'_{\mathbf{m}} &= (b_{m1}, b_{m2}, \ldots, b_{mm}, 0)\\
\mathbf{b}'_{\mathbf{m+1}} &= (t_1, t_2, \ldots, t_m, \beta),
\end{aligned}$$

where β is a parameter to be determined. Under some conditions, finding the shortest vector in this embedding lattice reveals the solution of the CVP instance.

Because LWE problem is an instance of BDD problem with target vector $\mathbf{v}$, we can utilize the embedding lattice (we called LWE-based lattice) to solve this problem. Estimating the λ_2-gap gives indication of the hardness of finding the shortest vector in this lattice.

To estimate the λ_2-gap for this lattice, we need to consider the low bound of λ_1 in the original q-ary lattice $\Lambda_q(A^T)$ and the length of the error $\mathbf{e}$. Now we use the estimation of integral points in high-dimension spheres to give the low bound of the minimum length in $\Lambda_q(A^T)$.

Lemma 3 *Let n, m be integers, and q be a prime such that $m > n$ and $q^{1-\frac{n+c}{m}} > \sqrt{\pi e}$ for some positive constant c. Uniformly choose matrix A from $\mathbb{Z}_q^{m\times n}$. Then with probability at least $1 - q^{-c}$, we have $\lambda_1(\Lambda_q(A^T)) \geq \min\{\sqrt{\frac{m}{2\pi e}} q^{1-\frac{n+c}{m}}, q\}$.*

Proof. The assumption $q^{1-\frac{n+c}{m}} > \sqrt{\pi e}$ implies $\sqrt{\frac{m}{2\pi e}} q^{1-\frac{n+c}{m}} \geq \sqrt{\frac{m}{2}}$. Let $R = \min\{\sqrt{\frac{m}{2\pi e}} q^{1-\frac{n+c}{m}}, q\}$. Then $N(m, R^2) \leq N(m, (\sqrt{\frac{m}{2\pi e}} q^{1-\frac{n+c}{m}})^2) \approx q^{m-n-c}$. Denote $V = \{\mathbf{u} \in \mathbb{Z}_q^m | \ \|\mathbf{u}\| \leq R\}$. Note that

$R \leq q$, we have $|V| = N(m, R^2)$. For uniformly random choice of A and any non-zero $\mathbf{s}$, since q is a prime, $Pr\{A\mathbf{s} = \mathbf{u}\} = q^{-m}$. The probability of existing a vector in $\Lambda_q(A^T) \cap V$ is

$$p = \Sigma_{\mathbf{s}\in\mathbb{Z}_q^n\setminus\mathbf{0}}\Sigma_{\mathbf{u}\in V} Pr\{A\mathbf{s} = \mathbf{u}\} \leq (N(m, R^2)/q^m)q^n \leq q^{-c}.$$

□

It is remarked that, references [11] and [37] give probabilistic low bounds for the minimum length in this lattice in l_∞ norm and Euclidean norm respectively. In [37], only the case $m \geq 5n \log q$ is considered, and the bound is $0.07\sqrt{m}q$ with probability $1 - q^{-n}$. For the same parameters and with the same probability, taking $c = n$ in our conclusion, we obtain a better bound $\sqrt{\frac{m}{2\pi e}}q^{1-\frac{n+c}{m}} = \frac{2^{-0.4}}{\sqrt{2\pi e}}\sqrt{m}q = 0.18\sqrt{m}q$.

As for the norm of the error vector, we need the tail bound on discrete Gaussians below:

Lemma 4 *[4] Let $c > 1, s > 0$ and n be a positive integer. Let $\mathbf{x} \in \mathbb{Z}^n$ be random and obey $D_{\mathbb{Z}^n,s}$. Then* $\Pr[\|\mathbf{x}\| \geq c\frac{s\sqrt{n}}{\sqrt{2\pi}}] \leq \left(c \cdot \exp(\frac{1-c^2}{2})\right)^n$.

It is remarked that $c \cdot \exp(\frac{1-c^2}{2}) < 1$ when $c > 1$. In fact, the function $f(c) = c \cdot \exp(\frac{-c^2}{2})$ is strictly decreasing on $[1, \infty)$.

From lemma 3 and 4, we can estimate the BDD factor $\gamma = \frac{\|e\|}{\lambda_1(\Lambda_q(A^T))}$ and use the reduction from $BDD_{\frac{1}{2\gamma}}$ to $uSVP_\gamma$ [18] to estimate the gap. However, owning to the special structure of LWE-based lattice and letting β be small, we can get the gap of this lattice directly without using the reduction and improve the result by a factor of 2.

Theorem 1. *Let n, m be integers, and q be a prime such that $m > n$ and $q^{1-\frac{n+c}{m}} > \sqrt{\pi e}$. $\mathbf{e}$ obeys $D_{\mathbb{Z}_q^m,\alpha q}$. Then for all but a fraction less than q^{-c+1} of the LWE-based lattice, we have the gap between λ_1 and λ_2 is larger than $\frac{R}{(1+\varepsilon)\alpha q\sqrt{\frac{m}{2\pi}}}$, where $R = \min\{\sqrt{\frac{m}{2\pi e}}q^{1-\frac{n+c}{m}}, q\}$.*

Proof. Denote $\mathbf{u} = A^T\mathbf{s}$. There exists a lattice vector $(\mathbf{v} - \mathbf{u}, \beta)$ with norm $\sqrt{\|\mathbf{e}\|^2 + \beta^2}$. Let $\beta \ll \|\mathbf{e}\|$, we have $\|(\mathbf{v} - \mathbf{u}, \beta)\| \approx \|\mathbf{e}\|$. Select $c = 1 + \varepsilon$, by lemma 4 and its remark, we know that $\|\mathbf{e}\| \leq (1 + \varepsilon)\alpha q\sqrt{\frac{m}{2\pi}}$ holds with overwhelming probability.

We will prove this vector is a unique shortest vector and there is a large λ_2-gap.

Write $\mathbf{y} \in \mathcal{L}'$ as $\mathbf{y} = \sum_{i=1}^{m+1} x_i\mathbf{b}'_i$. Partition the lattice points into sets $V_d = \{\mathbf{y} \in \mathcal{L}' | x_{m+1} = d\}$. Because of the characteristic of q-ary lattice, $V_{d_1} = V_{d_2}$ for $d_1 \equiv d_2 \ mod \ q$, so we just need to analyze the cases of $d = 1, 2, \ldots, q$. From lemma 3, we have the probabilistic lower bound for V_0. For other sets, same low bound can be obtained as follows:

Suppose there is a short vector $\mathbf{e}'$ with norm less than R which is in V_i. We ignore the last ordinate of $\mathbf{b}'_{m+1}$, $\mathbf{e}'$ and name them as $\mathbf{b}_{m+1}$, $\tilde{\mathbf{e}}$ respectively. Then we have the following equations: $i\mathbf{b}_{m+1} = A^T\mathbf{s} + \tilde{\mathbf{e}}$ and $\|\tilde{\mathbf{e}}\| < \|\mathbf{e}'\|$. For any $\mathbf{w}_j$ in $\Lambda_q^\perp(A^T)$ (q-dual lattice of $\Lambda_q(A^T)$), we have $\langle i\mathbf{b}_{m+1}, \mathbf{w}_j\rangle = \langle \mathbf{w}_j, \tilde{\mathbf{e}}\rangle \mod q$. Notice that there are only $m - n$ non-trivial linearly independent lattice vectors in $\Lambda_q^\perp(A^T)$. The short vector $\tilde{\mathbf{e}}$ has to satisfy $m - n$ equations in $\mathbb{Z}_q$. The randomness of matrix A implies this probability is $\frac{1}{q^{m-n}}$. So when $N(m, R^2)/q^{m-n} < q^{-c}$, with probability higher than $1 - q^{-c}$, V_i does not contain such short vector.

To sum up, for all the q partitions, the probability of no vector shorter than R is $(1 - q^{-c})^q \approx 1 - q^{-c+1}$. Since $\beta \ll (1 + \varepsilon)\alpha q\sqrt{\frac{m}{2\pi}}$, the gap between λ_1 and λ_2 is larger than $\frac{R}{(1+\varepsilon)\alpha q\sqrt{\frac{m}{2\pi}}}$, where $R = \min\{\sqrt{\frac{m}{2\pi e}}q^{1-\frac{n+c}{m}}, q\}$. □

As a consequence of this theorem, we can prove that a scheme based on LWE problem with one recommended parameters (see [11]) can be converted to find shortest vector in a sublattice with large λ_2-gap.

Corollary 1 *Consider the LWE problem with $m = 6n\log q$, prime $q \in [\frac{n^2}{2}, n^2]$, $\alpha = 1/(\sqrt{m}\cdot \log^2 m)$, and error distribution $D_{\mathbb{Z}_q^m,\alpha q}$. This problem can be attacked by a sublattice of embedding lattice of λ_2-gap bigger than $7.3\log^2 m$ with overwhelming probability.*

Proof. A simple obversion of LWE problem is that larger m can not increase its hardness. More precisely, for $A \in \mathbb{Z}_q^{m\times n}$, we take first m' rows of A to obtain $A' \in \mathbb{Z}_q^{m'\times n}$ ($m' \le m$), then construct corresponding embedding lattice denoted as $\mathcal{L}'_{m'}$. We aim at finding the optimal m' to maximize λ_2-gap in $\mathcal{L}'_{m'}$. Recovering the shortest vector of $\mathcal{L}'_{m'}$ can solve this LWE problem totally.

From theorem 1, $c = n$, $\sqrt{\frac{m'}{2\pi e}}q^{1-\frac{2n}{m'}}$ is an increasing function with m'. So when $\sqrt{\frac{m'}{2\pi e}}q^{1-\frac{2n}{m'}} < q$, λ_2-gap of $\mathcal{L}'_{m'}$ is larger than $\frac{1}{\sqrt{e}\alpha q^{\frac{2n}{m'}}}$ which increases with m'. When m' increases to make $\sqrt{\frac{m'}{2\pi e}}q^{1-\frac{2n}{m'}} > q$ hold, the low bound of λ_2-gap for $\mathcal{L}'_{m'}$ is $\frac{\sqrt{2\pi}}{\alpha\sqrt{m'}}$ which decreases with m'. Taking $\sqrt{\frac{m'}{2\pi e}} = q^{\frac{2n}{m'}}$, we obtain the optimal m'. For $m = 6n\log q, q \in [\frac{n^2}{2}, n^2]$, let $m' = 0.7n\log q = \frac{0.7m}{6}$, we have $\sqrt{\frac{m'}{2\pi e}}q^{1-\frac{2n}{m'}} > q$ as $n > 104$. The λ_2-gap in $\mathcal{L}'_{m'}$ is larger than $\frac{\sqrt{2\pi}}{\alpha\sqrt{m'}} \approx 7.3\log^2 m$ with probability close to 1. □

This result reveals that, the gap between λ_2 and λ_1 in LWE-based lattice is large.

Next we compare the efficiency of solving the LWE problem by searching the shortest vector in LWE-based lattice directly with the other two methods.

The first one is the distinguished attack proposed by Micciancio and Regev [24]. They solved the decision version of LWE by finding a short vector $\mathbf{w}$ in $\Lambda_q^{\perp}(A^T)$ and computing the inner product $\langle \mathbf{v}, \mathbf{w}\rangle$. If $\mathbf{v}$ is obtained from LWE instance, this inner product is distributed with the Gaussian distribution $D_{Z_q, s\|\mathbf{w}\|}$. And if $\mathbf{v}$ is chosen uniformly from $\mathbb{Z}_q^m$, this distribution is uniform. So when $\mathbf{w}$ is short enough to distinguish these two distribution, the attack succeeds. $exp(-\pi(\|\mathbf{w}\|\alpha)^2)$ is used to estimate the distinguished advantage. When the distinguished advantage is ε, α should be less than $\frac{1}{\|w\|}\sqrt{\frac{\ln(1/\varepsilon)}{\pi}}$. Using Hermit factor δ obtained from [8], $\alpha < \sqrt{\frac{ln(1/\varepsilon)}{\pi}}\delta^{-m}q^{-\frac{n}{m}}$ can be distinguished.

The other algorithm is the list-decoding algorithm proposed by Lindner and Peikert [20]. Their algorithm combines NearestPlane algorithm and enumeration algorithm to balance time/success tradeoff. The method uses lattice basis reduction to get a good basis in $\Lambda_q(A^T)$, and generalizes the NearestPlane algorithm to consider on some $d_i \ge 1$ distinct planes in the i-th level of the recursion. This technique makes the $\mathcal{P}_{1/2}(\tilde{\mathbf{B}})$ wider in the direction of $\tilde{\mathbf{b}}_i$ by a factor of d_i, where $\tilde{\mathbf{B}}$ denotes the set of Gram-Schmidt vectors. And $\|\tilde{\mathbf{b}}_i\|$ decreases sharply as i increases. The successful probability is $\Pi_{i=1}^m erf(\frac{d_i\|\tilde{b}_i\|\sqrt{\pi}}{2s})$. The number of enumeration is $\prod_{i=1}^n d_i$. According to GSA assumption and evaluation in [8], we have $\|\tilde{\mathbf{b}}_m\| = \frac{q}{\delta^m(det(\Lambda_q^{\perp}(A^T)))^{\frac{1}{m}}}$ and $\|\tilde{\mathbf{b}}_i\| = \|\mathbf{b}_1\|\alpha^{i-1}, 0 < \alpha < 1$. The value of erf function is less than 0.9 when the variable is smaller than 1.1. So $\alpha < \frac{d_i\sqrt{\pi}}{2.2}\delta^{-m}q^{-\frac{n}{m}}$ is efficient range for the attack. But the probability is the product for all $\tilde{\mathbf{b}}_i$.

Now, we analyze the security range of α for solving the BDD problem by embedding lattice. Just as our discussion in Corollary 1, $\sqrt{\frac{m}{2\pi e}}q^{1-\frac{n+c}{m}} > q$ holds, only when m is large. For LWE instance,

larger m is insecure. In the analysis of the two methods, $m = \sqrt{n \ln q / \ln \delta}$ minimizes $\delta^m q^{\frac{n}{m}}$, and obtains the optimal attack. As long as $n \ln q > 400$, $\delta > 1.005$ (based on the experiments in [8] Hermite factor less than 1.005 looks totally out of reach), $m = \sqrt{n \ln q / \ln \delta}$ makes $\sqrt{\frac{m}{2\pi e}} q^{1-\frac{n+c}{m}} < q$. So taking $R = \sqrt{\frac{m}{2\pi e}} q^{1-\frac{n+c}{m}}$ is reasonable. As mentioned in section 2, thanks to the conclusion in [8], our above analysis indicates that using the embedding lattice of LWE, we can solve the problem when $\frac{\sqrt{\frac{m}{2\pi e}} q^{1-\frac{n+c}{m}}}{(1+\varepsilon)\alpha q \sqrt{\frac{m}{2\pi}}} > \tilde{\delta}^m$, or more concisely $\alpha < \frac{1}{\sqrt{e}(1+\varepsilon)} \tilde{\delta}^{-m} q^{-\frac{n+c}{m}}$. This bound is not worse than the above two methods. Since we use the lower bound of gap, our analysis is conservative.

3.2 Reduction from BDD to uSVP Revisited

It is noted that LWE is a BDD instance. We assess the hardness of solving LWE by estimating the gap of its embedding lattice. More generally, using the embedding technique, BDD_{γ_1} instance can be reduced to $uSVP_\gamma$ problem. Note that $uSVP_\gamma$ refers to finding shortest vector when λ_2-gap is larger than γ. Thus, with larger λ_2-gap, solution to the BDD instance can be obtained easier. From the cryptanalysis point of view, fixing γ_1, the problem is easier if γ is large. In other words, fixing γ, larger γ_1 means better reduction.

The following elegant reduction is presented in [18]:

Lemma 5 *[18] For any constant $\gamma > 1$, there is a polynomial-time Cook reduction from $BDD_{\frac{1}{2\gamma}}$ to $uSVP_\gamma$.*

In this section, we discuss an improvement on this reduction. We assume that the distance between $\mathbf{t}$ and $\mathcal{L}$ is u. Using LLL reduced basis and NearestPlane algorithm [3], we get d satisfying $\frac{d}{2^n} < u \leq d$. By dividing the interval $(\frac{d}{2^n}, d]$ into polynomially many parts $(\frac{d}{2^n(1-\frac{1}{n})^i}, \frac{d}{2^n(1-\frac{1}{n})^{i+1}}]$, guessing $u \in (\frac{d}{2^n(1-\frac{1}{n})^{i_0}}, \frac{d}{2^n(1-\frac{1}{n})^{i_0+1}}]$, and taking $u_0 = \frac{d}{2^n(1-\frac{1}{n})^{i_0+1}}$, we have $u \leq u_0 < \frac{u}{1-\frac{1}{n}}$.

For the embedding lattice of $\mathcal{L}'$ mentioned above, we choose $\beta = ku_0$. The case of $k = 1$ corresponds to the reduction of $BDD_{\frac{1}{2\gamma}}$ to $uSVP_\gamma$ in [18]. By varing k, we can improve the reduction for $\gamma < 1.5321$.

Taking u_0 to be an approximation of u will not affect the reduction, more formally:

Lemma 6 *[18] For any constant c, there is a polynomial-time Cook reduction from BDD_α to $BDD_{\alpha(1-1/n)^c}$.*

Our main result of this subsection is follows:

Theorem 2. *For any $1 < \gamma < 1.5321$, there is a polynomial time Cook-reduction from BDD_{γ_1} to $uSVP_\gamma$ with $\gamma_1 = 1/(\sqrt{\frac{3\gamma^2}{4-\gamma^2}} + 1)$.*

Proof. If $\mathbf{v}$ is the closest lattice point to $\mathbf{t}$, we get the vector $(\mathbf{v} - \mathbf{t}, -ku_0) \in \mathcal{L}'$ with the norm $\sqrt{u^2 + k^2 u_0^2}$. If the vector is the $\gamma-$unique shortest vector in $\mathcal{L}'$, we can solve uSVP$_\gamma$ to get it. Suppose m is the coefficient of b'_{n+1}. Owning to the symmetry of lattice, we only consider $m \geq 0$. To guarantee the embedding lattice to be a u-SVP$_\gamma$ instance, the following three inequalities have to hold, the first condition comes from $m \geq 2$, the second is for $m = 0$, and the third refers to $m = 1$.

- $\|(\mathbf{v} - m\mathbf{t}, -mku_0)\| > \gamma\sqrt{u^2 + k^2 u_0^2}$ for all $m \geq 2$,

- $\lambda_1(L) > \gamma\sqrt{u^2 + k^2u_0^2}$,
- $\|(\mathbf{w} - \mathbf{t}, -ku_0)\| > \gamma\sqrt{u^2 + k^2u_0^2}$, if $\mathbf{w} \neq \mathbf{v}$.

We hope to get a larger γ_1 such that $u_0 < \gamma_1\lambda_1$, and in this case, $\mathrm{BDD}_{\gamma_1(1-\frac{1}{n})}$ can reduce to $uSVP_\gamma$.

Without loss of generality, we assume $2v \notin \mathcal{L}$. For the first constraint, we can get its sufficient condition such that

$$\|(\mathbf{v} - m\mathbf{t}, -mku_0)\| > 2ku_0 \geq \gamma\sqrt{u_0^2 + k^2u_0^2},$$

where $m \geq 2$. We obtain $k \geq \frac{\gamma}{\sqrt{4-\gamma^2}}$.

For the second constraint, when $\lambda_1(L) > \gamma\sqrt{u_0^2 + k^2u_0^2} \geq \gamma\sqrt{u^2 + k^2u_0^2}$, we can deduce that $u_0 < \frac{\lambda_1(L)}{\gamma\sqrt{k^2+1}}$ which implies γ_1 can be selected as $\frac{1}{\gamma\sqrt{k^2+1}}$.

For the third, we get a scope of γ_1 by the contradictory method. Provided that

$$\|(\mathbf{w} - \mathbf{t}, -ku_0)\| \leq \gamma\sqrt{u^2 + k^2u_0^2} \leq \gamma\sqrt{k^2 + 1}u_0.$$

It is easy to know, if $u_0 < \frac{\lambda_1(L)}{\sqrt{\gamma^2(k^2+1)-k^2}+1}$, we have

$$\|\mathbf{w} - \mathbf{v}\| \leq \|\mathbf{w} - \mathbf{t}\| + \|\mathbf{t} - \mathbf{v}\| \leq (\sqrt{\gamma^2(k^2 + 1) - k^2} + 1)u_0 < \lambda_1(L).$$

It is a contradiction. So, we get a scope of u_0 where the third constraint is true.

With k decreasing, γ_1 becomes larger. We take $k = \frac{\gamma}{\sqrt{4-\gamma^2}}$ to obtain $u_0 < \gamma_1\lambda_1$, here $\gamma_1 = \min\{\frac{\sqrt{4-\gamma^2}}{2\gamma}, 1/(\sqrt{\frac{3\gamma^2}{4-\gamma^2}} + 1)\}$. From $u \leq u_0 < \frac{u}{1-\frac{1}{n}}$, we finish the reduction of $BDD_{\gamma_1(1-\frac{1}{n})}$ to $uSVP_\gamma$. By lemma 6, we conclude that BDD_{γ_1} reduces to $uSVP_\gamma$.

It is easy to verify that, when $\gamma < 1.5321$, $\gamma_1 = 1/(\sqrt{\frac{3\gamma^2}{4-\gamma^2}} + 1) > 1/2\gamma$. □

Remark It is noted that if the factor of BDD is γ', then our reduction is to $uSVP_{\gamma''}$, with $\gamma'' = \frac{2(\frac{1}{\gamma'}-1)}{\sqrt{3+(\frac{1}{\gamma'}-1)^2}}$. This should be compared with that in [18], where $\gamma'' = \frac{1}{2\gamma'}$. When $0.5 > \gamma' > 0.3263$, our reduction implies a slightly larger λ_2-gap γ'', but do not exceed 1.5321.

Corollary 2 *There is a polynomial time Cook-reduction from $BDD_{\frac{\sqrt{3}}{2}}$ to SVP.*

Proof. If there are more than one closest lattice vectors for target $\mathbf{t}$, we only want to find one of them. So w.l.o.g we assume the closest vector is unique. When $\|\mathbf{w} - \mathbf{t}\| > \|\mathbf{v} - \mathbf{t}\|$, from the proof of theorem 2, the third condition $\|(\mathbf{w} - \mathbf{t}, -ku_0)\| > \gamma\sqrt{u^2 + k^2u_0^2}$ is trivial. So we only require $u_0 < \frac{\lambda_1(L)}{\gamma\sqrt{k^2+1}}$ with $k = \frac{\gamma}{\sqrt{4-\gamma^2}} = \frac{\sqrt{3}}{3}$. This proves that $BDD_{\frac{\sqrt{3}}{2}}$ reduces to SVP. □

Remark We present the fact that when the distance between the target vector and the lattice is smaller than $\frac{\sqrt{3}}{2}\lambda_1$, we can capture the closest lattice vector by solving one SVP instance. The reduction of $\mathrm{BDD}_{\frac{1}{2\gamma}}$ to uSVP_γ of [18] only implies the reduction of $BDD_{\frac{1}{2}}$ to SVP. Our result improves the parameter from $\frac{1}{2}$ to $\frac{\sqrt{3}}{2}$.

4 Reduced-Dimension Algorithm for SVP of Lattices with $\lambda_{\varepsilon n+1}$-Gap

In this section, our purpose is to obtain the shortest vector of lattice with $\lambda_{\varepsilon n+1}$-gap by utilizing a modified sieve for approximate SVP and the best known SVP algorithm [26] to find the shortest vector in a reduced-dimension sublattice. The presence of $\lambda_{\varepsilon n+1}$-gap makes it possible to sieve enough c-approximate shortest vectors which distribute in a reduced-dimension lattice.

4.1 Sieve Algorithm for $\mathbf{SVP}_\gamma$

We modify ListSieve-Birthday[31] sieve to an approximate SVP algorithm (see algorithm 1) to sieve sufficiently many c-approximate shortest vectors. Compared with ListSieve-Birthday sieve, the algorithm terminates the sieve early and relaxes the birthday search. The number N_2 of sieved vectors is much smaller than that of ListSieve-Birthday algorithm, which reduces the time complexity significantly.

Sample Algorithm

Input: An LLL-reduced basis of a lattice L, perturbation radius $\xi\mu$,where $\xi > \frac{1}{2}$
Output: A lattice vector $\mathbf{u}$ and a perturbed vector $\mathbf{u}'$
1: Choose $\mathbf{x}$ randomly in $B_n(\xi\mu)$
2: $\mathbf{u}' \longleftarrow (-\mathbf{x}) \mod \mathcal{P}(B)$
3: $\mathbf{u} \longleftarrow \mathbf{u}' + \mathbf{x}$
4: Return $(\mathbf{u}, \mathbf{u}')$

Reduction Algorithm

Input: A pair $(\mathbf{u}, \mathbf{u}')$ and a list $T \subseteq L$, reduced factor $\delta < 1$
Output: A reduced pair $(\mathbf{u}, \mathbf{u}')$
1: While $(\exists \mathbf{w} \in T) : \|\mathbf{u}' - \mathbf{w}\| \leq \delta\|\mathbf{u}'\|$
2: $\quad (\mathbf{u}, \mathbf{u}') \longleftarrow (\mathbf{u} - \mathbf{w}, \mathbf{u}' - \mathbf{w})$
3: end while
4: Return $(\mathbf{u}, \mathbf{u}')$

The complexity estimation of the approximation SVP algorithm is based on the following lemmas.

Lemma 7 *[31] Let $c_l(c, \xi) = -\frac{1}{2}\log_2(1 - \frac{2\xi}{c}) + 0.401$. The list T in the algorithm contains at most $N_L(n) = 2^{c_l n + o(n)}$ vectors.*

Lemma 8 *[28] Let $c_g(\xi) = -\frac{1}{2}\log_2(1 - \frac{1}{4\xi^2})$, and $\mathbf{s}$ be a shortest non-zero vector of $\mathcal{L}(\mathbf{B})$. $I_s = \{\mathbf{x} \in B_n(\xi u) : \mathbf{x} + \mathbf{s} \in B_n(\xi u)\}$. If $\mathbf{x}$ is chosen uniformly in $B_n(\xi u)$, then $Pr(\mathbf{x} \in I_s) \geq \frac{1}{N_G}$ where $N_G = 2^{c_g n + o(n)}$.*

Lemma 9 *[31] Let $N_1 = 4\lceil N_L N_G \rceil$, $p_i = Pr(E_i | \mathbf{x}_i \in I_s)$, where E_i is the event $\|\mathbf{u}_i\| < c\mu$. Set $J = \{i \leq N_1 : p_i \leq \frac{1}{2}\}$. Then $|J| \leq N_1/2$.*

Algorithm 1: The Approximate SVP Algorithm

Input: A basis B, $\frac{c}{2} > \xi > \frac{1}{2}$, N_1,N_2,$c > 1$, $d > 1$, dimension n, reduced factor $\delta < 1$,$\mu \simeq \lambda_1$
Output: d different lattice vectors with norm less than $c\mu$ or a shortest lattice vector
1: $T \longleftarrow \emptyset$,$U \longleftarrow \emptyset$,$U' \longleftarrow \emptyset$
2: For $i = 1$ to N_1 do
3: $(\mathbf{t}_i, \mathbf{t}'_i) \longleftarrow$ Reduction(Sample(B,$\xi\mu$),T,δ)
4: If $\|\mathbf{t}_i\| \geq c\mu$ then
5: $T \longleftarrow T \cup \{\mathbf{t}_i\}$
6: end if
7: end for
8: For $i = 1$ to N_2 do
9: $(\mathbf{u}_i, \mathbf{u}'_i) \longleftarrow$ Reduction(Sample(B,$\xi\mu$),T,δ)
10: $U \longleftarrow U \cup \{\mathbf{u}_i\}$
11: if $\|\mathbf{u}_i\| \leq c\mu$ then
12: $U' \longleftarrow U' \cup \{\mathbf{u}_i\}$
13: end for
14: if $(|U' \setminus \{0\}| \geq d)$ then
15: Return U'
16: else find closest distinct points $(\mathbf{u}_i, \mathbf{u}_j)$ in U
17: Return $\mathbf{u}_i - \mathbf{u}_j$

The previous lemma implies that when the number of points chosen to construct the list is large enough, then with high probability, the new vector can be reduced to a lattice vector shorter than $c\mu$. In the second loop of the algorithm 1, the points are not added to the list. Both of the loops use the same procedure to sieve points. So this lemma still holds for the second loop. From this, we get the next lemma whose proof is very similar to that of lemma 7 in [31], and it is provided in the appendix.

Lemma 10 *Let $N_2 = 22dN_G$. Assume that n is sufficiently large. Then with probability higher than $\frac{1}{8}$, either there exist d different vectors shorter than $c\mu$, or the shortest vector is returned .*

Now, we are able to evaluate the complexity of algorithm 1.

Theorem 3. *Let $c_{time} = \max\{2c_l(c,\xi) + c_g(\xi), 2c_g\}$ and $c_{space} = \max\{c_l(c,\xi), c_g(\xi)\}$, when $d = o(2^{c_g n})$. Then with probability not less than $\frac{1}{8}$, the algorithm 1 returns d non-zero c-SVP vectors or a shortest vector with time $\tilde{O}(2^{c_{time}n+o(n)})$ and space $\tilde{O}(2^{c_{space}n+o(n)})$. In particular, when $\xi = \frac{\sqrt[3]{c}}{2}$, time complexity is optimal: $c_{time} = 0.802 - 1.5\log_2(1 - c^{-\frac{2}{3}})$ and $c_{space} = 0.401 - 0.5\log_2(1 - c^{-\frac{2}{3}})$*

Proof. The time complexity of first loop in steps 2-7 is $N_1 N_L$, and the second loop of steps 8-13 is $N_2 N_L$. If we have to implement step 16, the complexity of steps 14-17 is at most N_2^2. Otherwise this part is negligible. So the total time complexity is $\tilde{O}(2^{c_{time}n+o(n)})$ as $c_{time} = \max\{2c_l(c,\xi) + c_g(\xi), 2c_g\}$. It is obvious that space complexity is $|T|+|U| = \tilde{O}(2^{c_{space}n+o(n)})$, as $c_{space} = max\{c_l(c,\xi), c_g(\xi)\}$.

Plugging $\xi = \frac{\sqrt[3]{c}}{2}$ into $c_{time} = 2c_l(c,\xi) + c_g(\xi) = 0.802 - \log_2((1 - \frac{2\xi}{c})\sqrt{1 - \frac{1}{4\xi^2}})$, we get the minimum of time complexity, $c_{time} = 0.802 - 1.5\log_2(1 - c^{-\frac{2}{3}})$. The corresponding space complexity is $c_{space} = 0.401 - 0.5\log_2(1 - c^{-\frac{2}{3}})$. □

According to Theorem 3, for a fixed c, in table 1, the columns marked "Algorithm 1" list some optimal time complexity and corresponding ξ and space complexity. We see that when c

gets larger, the complexity decreases obviously. In particular, taking c as a function of n such that $\lim_{n\to\infty} c(n) = \infty$, then we can find the approximate shortest vector in time $\tilde{O}(2^{0.802n+o(n)})$ and space $\tilde{O}(2^{0.401n+o(n)})$.

If we are just interested in solving approximate SVP, then it is sufficient to find a non-zero vector that is shorter than $c\mu$. The algorithm 1 can be modified as follows:

In step 4, after the reduction steps, if $\|\mathbf{t}_i\| > c\mu$, we check all the $\mathbf{t}_j \in T$. If there exists a $\mathbf{t}_j$, s.t. $\|\mathbf{t}_i - \mathbf{t}_j\| < c\mu$, the algorithm terminates. Otherwise, we add $\mathbf{t}_i$ to the list. We now argue that this $\mathbf{t}_i - \mathbf{t}_j$ is a non-zero vector. In fact, $(\mathbf{t}_i, \mathbf{t}'_i)$ is obtained by the Reduction Algorithm, so $\|\mathbf{t}'_i - \mathbf{t}_j\| > (1 - \frac{1}{n})\|\mathbf{t}'_i\|$ holds (taking $\delta = 1 - \frac{1}{n}$). Suppose $\mathbf{t}_i - \mathbf{t}_j = 0$, when n is large, we have

$$\|\mathbf{t}_i\| \leq \|\mathbf{x}_i\| + \|\mathbf{t}'_i\| < \xi\mu + \frac{\|\mathbf{t}'_i - \mathbf{t}_j\|}{1 - \frac{1}{n}} = \xi\mu + \frac{\|\mathbf{t}'_i - \mathbf{t}_i\|}{1 - \frac{1}{n}} < \xi\mu + \frac{\|\mathbf{x}_i\|}{1 - \frac{1}{n}} < c\mu$$

This is a contradiction. This implies that when the algorithm terminates at this step, a short vector whose length is less than $c\mu$ is obtained. When the first loop ends, if the algorithm doesn't terminate, then for all the points in T, we must have $\|\mathbf{t}_i - \mathbf{t}_j\| > c\mu$. Using an analysis that is similar to that in [25] (we give the detailed proof in appendix) we get a smaller $c_l = \log_2 \frac{\sqrt{\xi^2+c^2}+\xi}{c} + 0.401$.

Lemma 11 *Let $c_l = \log_2 \frac{(\sqrt{\xi^2+c^2}+\xi)}{c} + 0.401$ in our modified c-SVP algorithm. During the process of the algorithm, the List T contains at most $N_L(n) = 2^{c_l n+o(n)}$ vectors.*

Adding these checks will not affect the upper bound of time complexity. Other parts of the estimation of complexity are the same as lemmas 8-10 and theorem 4. For a fixed c, we list some optimal time complexity in columns marked "Modified Algorithm" in table 1. This algorithm also have time $\tilde{O}(2^{0.802n+o(n)})$ and space $\tilde{O}(2^{0.401n+o(n)})$, as long as $\lim_{n\to\infty} c(n) = \infty$

	Algorithm 1			Modified Algorithm		
c	ξ	time	space	ξ	time	space
2.71	–	–	–	0.8201	1.9976	0.8316
3.61	0.7670	1.9993	0.8001	0.8810	1.7798	0.7497
8	1	1.4246	0.6085	1.0810	1.3643	0.5954
15	1.2331	1.1907	0.5306	1.3010	1.1672	0.5260

Table 1.

In our main algorithm in next subsection, this improvement helps to treat the case when there is a gap between λ_2 and λ_1. Because in that case, only one short vector is sufficient to find the shortest one.

It is remarked that when $c < n^\gamma$ with $\gamma \approx 1.5$, our algorithm is better than the slide reduction [9] which is believed to be the best known approximate algorithm.

4.2 Reduced-Dimension Algorithm of SVP for Lattices with $\lambda_{\varepsilon n+1}$-Gap

If $\lambda_{\varepsilon n+1} > c\lambda_1$, $\varepsilon < 1$, we can use the sieve to get enough lattice points in $B_n(c\lambda_1)$ which also belong to a sublattice $span\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{\varepsilon n}\} \cap \mathcal{L}(B)$ of dimension εn, where $\|\mathbf{s}_i\| = \lambda_i$. So solving the SVP of the original $\mathcal{L}$ can be reduced to SVP of the reduced-dimension sublattice.

Algorithm 2 is our reduced-dimension algorithm for searching shortest lattice vectors in the presence of $\lambda_{\varepsilon n+1}$-gap. The sieve in algorithm 1 collects a set of short vectors with norm less than $c\lambda_1$. Because of the $\lambda_{\varepsilon n+1}$-gap, these vectors generate a sublattice included the shortest vector in the original lattice. In this case, we reduce the n-dimensional SVP problem to that of dimension εn and cut the time complexity obviously. This algorithm consists of three parts. First part sieves enough short vectors which fall into a subspace. The core of the second part is to apply lemma 2 and to output a basis of a sublattice whose shortest vector is the same as the original one. The final part finds the shortest lattice vector in the reduced-dimension sublattice by the Deterministic SVP Algorithm of [26].

Algorithm 2: Reduced-Dimension Algorithm for lattice with $\lambda_{\varepsilon n+1}$-Gap

Input: A basis of n-dimension lattice $\mathbf{B}$, $\frac{c}{2} > \xi > \frac{1}{2}$, N_1,N_2,$c > 1$, $\varepsilon < 1$, reduced factor $\delta < 1$,$\mu \simeq \lambda_1$
Output: A shortest lattice vector
1: $U \longleftarrow \emptyset$
2: $U \longleftarrow$ Approximate SVP Algorithm$(\mathbf{B}, \xi, N_1, N_2, c, \varepsilon^2 n^2, n, \delta, \mu)$
3: $\tilde{\mathbf{B}} \longleftarrow$ SubLattice Algorithm$(\mathbf{B}, U, \varepsilon n)$
4: $\mathbf{s} \longleftarrow$ Deterministic SVP Algorithm$(\tilde{\mathbf{B}}, \varepsilon n)$

In the following lemma, we prove that the shortest vector belongs to the sublattice generated by the short vectors we obtained from algorithm 1 by taking $d = \varepsilon^2 n^2$. This conclusion guarantees that we will not lose the shortest vector even though we only consider the reduced-dimension lattice.

Lemma 12 *For the $\varepsilon^2 n^2$ different non-zero lattice vectors in set U, with probability close to 1, the shortest vector $\mathbf{s} \in span(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{\varepsilon^2 n^2})$.*

Proof. Let $H_i = span(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{i\varepsilon n})$, $i = 1, 2, \ldots, \varepsilon n$. If $dim(H_i) < dim(H_{i+1})$ hold for all $i = 1, \ldots \varepsilon n - 1$, $dim(H_{\varepsilon n}) = dim(span(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{\varepsilon^2 n^2})) = \varepsilon n$. It is obvious that $\mathbf{s} \in H_{\varepsilon n}$.

If there exist i_0 ($i_0 < \varepsilon n$), s.t. $dim(H_{i_0}) = dim(H_{i_0+1})$, we prove that $\mathbf{s} \in H_{i_0}$ with probability higher than $1 - (\frac{1}{2})^{\varepsilon n}$.

Suppose $\mathbf{s} \notin H_{i_0}$. Since $N_2 = 22d2^{c_g n}$, we consider those $\mathbf{u}_i$ satisfying $\mathbf{u}_i = \mathbf{u}'_i + \mathbf{x}_i$, where $\mathbf{x_i} \in I_s$. If the perturbation $\mathbf{x} \in I_s$, in Sample Algorithm we have,

$$\mathbf{v}'_x = -\mathbf{x} \mod \mathcal{P}(B) = (-\mathbf{x} + \mathbf{s}) \mod \mathcal{P}(B) = \mathbf{v}'_{-\mathbf{x}+\mathbf{s}}.$$

According to List-Birthday Sieve, algorithm 1 implies that $Pr(\mathbf{u} = \mathbf{v}) = Pr(\mathbf{u} = \mathbf{v} + \mathbf{s})$. For any $\mathbf{v} \in H_{i_0}$, from $\mathbf{s} \notin H_{i_0}$ we get $\mathbf{v} + \mathbf{s} \notin H_{i_0}$. So,

$$\begin{aligned} Pr(\mathbf{u} \in H_{i_0}) &= \Sigma_{\mathbf{v} \in H_{i_0}} Pr(\mathbf{u} = \mathbf{v}) = \Sigma_{\mathbf{v} \in H_{i_0}} Pr(\mathbf{u} = \mathbf{v} + \mathbf{s}) \\ &\leq \Sigma_{\mathbf{v} \notin H_{i_0}} Pr(\mathbf{u} = \mathbf{v}) = Pr(\mathbf{u} \notin H_{i_0}) \end{aligned} \quad (1)$$

which implies $Pr(\mathbf{u} \in H_{i_0}) \leq \frac{1}{2}$. If $dim(H_{i_0}) = dim(H_{i_0+1})$, we have $\mathbf{u}_i \in H_{i_0}$ for all $i = i_0 \varepsilon n + 1, \ldots, (i_0 + 1)\varepsilon n$. This happens with probability less than $(\frac{1}{2})^{\varepsilon n}$. Thus $\mathbf{s} \in H_{\varepsilon n}$ holds with the probability close to 1 □

Without loss of generality, we assume the dimension of $span(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{\varepsilon^2 n^2})$ is εn in the analysis below.

Theorem 4. *When $\lambda_{\varepsilon n+1} > c\lambda_1$, the time complexity of algorithm 2 is $\tilde{O}(2^{c_{time}n+o(n)})$, where $c_{time}(c,\xi,\varepsilon) = max\{2c_l(c,\xi)+c_g(\xi), 2\varepsilon n\}$, and space complexity is $\tilde{O}(2^{c_{space}n+o(n)})$ with $c_{space}(c,\xi,\varepsilon) = max\{c_l(c,\xi), c_g(\xi), \varepsilon n\}$.*

Proof. We have proved in theorem 3, the time complexity of step 2 of Reduced-Dimension Algorithm is $\tilde{O}(2^{c_{time_1}n+o(n)})$ where $c_{time_1}(c,\xi) = 2c_l(c,\xi) + c_g(\xi)$. And with $N_2 = 22d2^{c_g n}$, $\tilde{O}(2^{c_g n+o(n)})$ is sufficient for step 3. For step 4, the shortest vector for lattice with dimension εn can be found within $\tilde{O}(2^{2\varepsilon n+o(n)})$ time. For step 1, $\tilde{O}(2^{c_{space_1}n+o(n)})$ is needed when $c_{space_1}(c,\xi) = max\{c_l(c,\xi), c_g(\xi)\}$. Step 3 and Step 4 cost space $\tilde{O}(2^{c_g n+o(n)})$ and $\tilde{O}(2^{\varepsilon n+o(n)})$ respectively. Putting the three parts together, we conclude the theorem. □

4.3 Reduced-Dimension Algorithm for SVP of Lattices with Multi-Gap

If a lattice has multi-gap among successive minima, more precisely, there exist $\varepsilon_1 n$, $\varepsilon_2 n$,...$\varepsilon_t n$ such that $\lambda_{\varepsilon_1 n+1} > c_1\lambda_1$, $\lambda_{\varepsilon_2 n+1} > c_2\lambda_1$, $\cdots$, $\lambda_{\varepsilon_t n+1} > c_t\lambda_1$, where $\varepsilon_{i-1} < \varepsilon_i$, $c_{i-1} < c_i$, we explore a recursive algorithm based on the algorithm 2. We take ε_i to be the smallest value that makes $\lambda_{\varepsilon_i n+1} > c_i\lambda_1$. We have discussed the case of $t = 1$ in Section 4.2. To treat the general case, set $\varepsilon_{t+1} = 1$, $\tilde{\mathbf{B}}_0 = \mathbf{B}$ in algorithm 3.

Algorithm 3: Multi-Gap lattice Algorithm

Input: A basis of n-dimension lattice $\mathbf{B}$,$(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_t)$ and $(c_1, c_2, \ldots, c_t)$,reduced factor $\delta < 1$,$\mu \simeq \lambda_1$

Output: A shortest lattice vector

1:$U \longleftarrow \emptyset$
2: For $i = 0$ to $i = t-1$
3: $U_i \longleftarrow$ Approximate SVP Algorithm$(\tilde{\mathbf{B}}_\mathbf{i}, \xi^i, N_1^i, N_2^i, c_{t-i}, \varepsilon_{t-i}^2 n^2, \varepsilon_{t-i+1} n, \delta, \mu)$
4: $\tilde{\mathbf{B}}_{i+1} \longleftarrow$SubLattice Algorithm$(\tilde{\mathbf{B}}_\mathbf{i}, U_i, \varepsilon_{t-i} n)$
5: end for
6: $\mathbf{s} \longleftarrow$ Deterministic SVP Algorithm$(\tilde{\mathbf{B}}_\mathbf{t}, \varepsilon_1 n)$

Denote the time complexity of algorithm 1 for n-dimension lattice as $\tilde{O}(2^{c_{time1}(c,\xi)n+o(n)})$ and $\tilde{O}(2^{c_{space1}(c,\xi)n+o(n)})$ for space. The total time and space complexity is $\tilde{O}(2^{c_{time}+o(n)})$ and $\tilde{O}(2^{c_{space}+o(n)})$ respectively where

$$c_{time} = max\{c_{time1}(c_t, \xi^0)n, c_{time1}(c_{t-1}, \xi^1)\varepsilon_t n, \ldots, c_{time1}(c_1, \xi^{t-1})\varepsilon_2 n, 2\varepsilon_1 n\},$$

$$c_{space} = max\{c_{space1}(c_t, \xi^0)n, c_{space1}(c_{t-1}, \xi^1)\varepsilon_t n, \ldots, c_{space1}(c_1, \xi^{t-1})\varepsilon_2 n, \varepsilon_1 n\}.$$

Since the complexity of Deterministic SVP Algorithm is $\tilde{O}(2^{2n+o(n)})$, the key parameter is c_0 which is minimal c satisfying $c_{time1}(c,\xi) < 2$. Based on our analysis above, c_0 is 3.61. For a lattice $\lambda_{n-o(n)} < c_0\lambda_1$, the Deterministic SVP Algorithm performs better. We can consider the gaps between this kind of lattices too small to be used to simplify the general algorithm for SVP.

Note that the Deterministic Algorithm in [26] can solve the Successive Minima Problem in the same time $\tilde{O}(2^{2n+o(n)})$. If the dimension of sublattice obtained is $\varepsilon_1 n$, and in step 6 of algorithm 3, by invoking the deterministic algorithm to solve SMP in this sublattice. We can generalize our algorithm to find part of the Successive Minima Problem for the multi-gap lattice with the same complexity as solving SVP in this lattice.

5 Conclusion

In this paper, we first estimate the size of the λ_2-gaps in embedding lattice obtained from LWE problem. Solving SVP in this lattice would reveal the solution of corresponding LWE problem. Based on this evaluation, our analysis shows that the efficiency of attacking LWE problem by embedding lattice directly is almost the same as distinguish attack and list decoding attack. For general BDD problem, we present a better reduction from BDD_{γ_1} to $uSVP_{\gamma}$ with larger λ_2-gap for some value of γ_1

We also give a more efficient algorithm to solve SVP of lattices with gaps among the successive minima. The key of the algorithm is reducing SVP of a lattice to the SVP of a reduced-dimension sublattice. Especially for the lattices with multi-gap, our algorithm can be invoked recursively.

References

1. Ajtai,M.,Dwork,C.: A public-key cryptosystem with worst-case/average-case equivalence. In STOC 1997. PP.284-293. ACM, New York(1997).
2. Ajtai,M., Kumar,R., Sivakumar,D.: A sieve algorithm for the shortest lattice vector problem. In: STOC 2001. pp. 266-275.ACM, New York(2001).
3. Babai,L.: On lovasz' lattice reduction and the nearest lattice point problem. Combinatorica 6(1),1-13(1986).
4. Banaszczyk,W.: New bounds in some transference theorems in the geometry of numbers. Mathematische Annalen, 296(4),625-635(1993).
5. Coster,M.J., Joux,A., La Macchia,B.A., Odlyzko,A.M., Schnorr,C.P., Stern,J.: An improved low-density subset sum algorithm, Computational Complexity.2 , 97-186(1992).
6. Coppersmith, D., Shamir,A.:Lattice attacks on NTRU. In: EUROCRYPT 1997. LNCS, vol.1233, pp.52-61. Springer, Heidelberg(1997)
7. Conway,J., Sloane,N.: Sphere Packing, Lattices and Groups. Springer-Verlag.Third edition(1998).
8. Gama,N., Nguyen,P.Q.: Predicting Lattice Reduction. In: EUROCRYPT 2008. LNCS,vol. 4965, pp.31-51.Springer, Heidelberg(2008).
9. Gama,N., Nguyen,P.Q.: Finding Short Lattice Vectors within Mordell's Inequality. In: STOC 2008. pp.207-216. ACM, New York(2008).
10. Gama,N., Nguyen,P. Q., Regev,O.: Lattice Enumeration Using Extreme Pruning. In EUROCRYPT 2010. LNCS, vol.6110, pp. 257-278. Springer, Heidelberg(2010).
11. Gentry, C. , Peikert, C. , Vaikuntanathan,V.: Trapdoors for hard lattices and new cryptographic constructions. In STOC 2008. pp. 197-206. ACM, New York(2008).
12. Han,D., Kim, M-H., Yeom,Y.: Cryptanalysis of the Paeng-Jung-Ha cryptosystem from PKC 2003. In PKC 2007. LNCS,vol. 4450,pp. 107-117. Springer, Heidelberg(2007).
13. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A Ring-Based Public Key Cryptosystem. In: ANTS 1998. LNCS, vol. 1423, pp. 267-288. Springer, Heidelberg (1998)
14. Kannan,R.: Improved algorithms for integer program ming and related lattice problems. In STOC 1983. pp.193-206. ACM, New York(1983).
15. Kannan,R.: Minkowski's convex body theorem and integer programming. Mathematics of Operations Research,12(3),415-440(1987).
16. G. Kabatiansky and V. Levenshtein. Bounds for packings on a sphere and in space. Problemy Peredachi Informatsii, 14(1):3-25, 1978.
17. Lenstra,A. K., Lenstra, H. W.Jr., Lovász,L.: Factoring polynomials with rational coefficients. Mathematische Annalen 261,513-534(1982).
18. Lyubashevsky,V., Micciancio,D.: On Bounded Distance Decoding, Unique Shortest Vectors, and the Minnimum Distance Problem. In CRYPTO 2009. LNCS, vol.5677, pp.577-594. Springer,Heidelberg(2009).
19. Lagarias,J.C.,Odlyzko, A.M.: Solving low-density subset sum problems. Jounal of the Association for Computing Machinery, 32(1): 229-246(1985).
20. R.Lindner and C.Peikert. Better key sizes (and attacks) for LWE-based encryption.In: Cryptology ePrint Archive, Report 2010/613(2010).

21. Micciancio,D.: Efficient Reductions among Lattice Problems. In SODA 2008. pp. 84-93. ACM/SIAM, New York(2008).
22. Milnor,J.,Husemoller,D.: Symmetric Bilinear Forms. Math.Z(1973).
23. Mazo,J.E., Odlyzko,A.M.: Lattice points in high-dimensional sheres. Monatsh. Math.,110:47-61(1990).
24. Micciancio,D., Regev.O,: Post-Quantum Cryptography, chapter Lattice-based Cryptography. Springer-Verlag(2008).
25. Micciancio,D., Voulgaris,P.:Faster exponential time algorithms for the shortest vector problem. In SODA 2010.pp. 1468-1480, ACM/SIAM, New York(2010).
26. Micciancio,D., Voulgaris,P.: A Deterministic Single Exponential Time Algorithm for Most Lattice Problems based on Voronoi Cell Computations. In STOC 2010. pp.351-358. ACM, New York(2010).
27. Nguyen,P.Q.:Cryptanalysis of the Goldreich-Goldwasser-Halevi cryptosystem from Crypto'97. In CRYPTO 1999. LNCS, vol.1666, pp. 288-304. Springer, Heidelberg(1999).
28. Nguyen,P.Q., Vidick,T.: Sieve algorithms for the shortest vector problem are practical. J. of Mathematical Cryptology, 2(2),181-207(2008).
29. C.Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In STOC 2009. pp.333-342. ACM, New York(2009).
30. Pohst,M.: On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications. ACM SIGSAM Bulletin 15(1),37-44(1981).
31. Pujol,X. and Stehlé,D.: Solving the shortest lattice vector problem in time $2^{2.465n}$. Cryptology ePrint Archive, Report 2009/605(2009).
32. Plantard,T., Susilo,W.: Broadcast attacks against lattice-based cryptosystems. In ACNS 2009. LNCS,vol.5536, pp. 456-472. Springer, Heidelberg(2009).
33. Regev.O.: Lecture notes on lattices in computer science, 2004. Available at http://www.cs.tau.ac.il/odedr/teaching/lattices fall 2004/index. html.
34. Regev.O: On lattices, learing with errors, random linear codes, and cryptography. J.ACM 56(6),1-40(2009).
35. Schnorr,C.P.: A hierarchy of Polynomial Lattice Basisi Reduction Algorithms. Theoretical Computer Science, 53,201-224(1987).
36. Schnorr,C. P., Euchner,M.: Lattice basis reduction: improved practical algorithms and solving subset sum problems. Mathematics of Programming 66,181-199(1994).
37. Stehlé,D., Steinfeld,R., Tanaka,K., Xagawa,K.: Efficient public key encryption based on ideal lattices. In: ASIACRYPT 2009. LNCS, vol.5912, pp.617-635. Springer, Heidelberg(2009).
38. Wang, X.Y, Liu, M.J, Tian, C.L, Bi,J.G. Improved Nguyen-Vidick Heuristic Sieve Algorithm for Shortest Vector Problem. In : ASIACCS 2011, pp 1-9,2011

A Appendix

Proof of Lemma 10 Let $\mathbf{s}$ be the shortest vector of the lattice whose norm is approximately μ, and record $I_s = \{\mathbf{x} \in B_n(\xi u), \mathbf{x}+\mathbf{s} \in B_n(\xi u)\}$. Note that the distribution of the vectors in list U are iid and $\mathbf{u}_i - \mathbf{u}'_i = \mathbf{x}_i$ is preserved during the sieve process. In the second loop of algorithm 2, reduction process is the same as in the first loop. According to lemma 9, for more than half of the point, $Pr(\|\mathbf{u}_i\| \leq c\mu | \mathbf{x}_i \in I_s) \geq \frac{1}{2}$ and we have

$$Pr((\|\mathbf{u}_i\| \leq c\mu) \cap (\mathbf{x}_i \in I_s)) = Pr(\|\mathbf{u}_i\| \leq c\mu | \mathbf{x}_i \in I_s) Pr(\mathbf{x}_i \in I_s) \geq \frac{1}{2N_G}.$$

Let $A = |\{\mathbf{u}_i | Pr((\|\mathbf{u}_i\| \leq c\mu) \cap (\mathbf{x}_i \in I_s)) \geq \frac{1}{2N_G}\}|$. Based on the analysis above, the variable A has a binomial distribution of parameter $p \geq \frac{1}{2}$. So $\mathbb{E}(A) = pN_2 \geq N_2/2$ and $\mathbb{D}(A) = p(1-p)N_2$. By Chebyshev's inequality we have

$$Pr(A \leq \frac{3N_2}{8}) \leq Pr(|A - \mathbb{E}(A)| \geq \mathbb{E}(A) - \frac{3N_2}{8}) \leq \frac{\mathbb{D}(A)}{(\mathbb{E}(A) - \frac{3N_2}{8})^2} \leq \frac{16}{N_2} \leq \frac{1}{2}$$

when $N_2 \geq 32$.

We define $X = \{i \leq N_2 : (\|\mathbf{u}_i\| \leq c\mu) \cap (\mathbf{x}_i \in I_s)\}$, using the Chebyshev's inequality again, we have $|X| \geq d$ with probability higher than $\frac{1}{2}$. If these d vectors are different, our algorithm stops. Next we prove that collision means the occurrence of the shortest vector with high probability.

Now let us restrict to the vectors in X and prove the conclusion by considering the conceptual modification τ which first introduced by Regev [33]. Since in our algorithm, sometimes $\xi > 1$, and $I_s \cap (I_s + \mathbf{s})$ is not empty. We modify the definition of τ in the following manner. Let $\tau : B_n(\xi u) \longmapsto B_n(\xi u)$ be such that $\tau(\mathbf{x}) = \mathbf{x}+\mathbf{s}$ if $x \in I_s \setminus (I_s+\mathbf{s})$, $\tau(\mathbf{x}) = \mathbf{x}-\mathbf{s}$ if $x \in (I_s+\mathbf{s}) \setminus I_s$, and $\tau(\mathbf{x}) = \mathbf{x}+\mathbf{s}$ with probability $\frac{1}{2}$, $\tau(\mathbf{x}) = \mathbf{x} - \mathbf{s}$ with probability $\frac{1}{2}$, if $\mathbf{x}_i \in I_s \cap (I_s + \mathbf{s})$. τ is identity elsewhere. We modify the Sample Algorithm in the second loop by applying τ with probability $1/2$ on every perturbation $\mathbf{x}$. Note that

$$\mathbf{u}'_x = -\mathbf{x} \mod \mathcal{P}(B) \quad = \quad \tau(-\mathbf{x}) \mod \mathcal{P}(B) \quad = \quad \mathbf{u}'_{\tau(-x)},$$

and the reduction step in the second loop sieve $\mathbf{u}'$ rather than $\mathbf{u}$. Furthermore, τ maintains the uniform distribution on $B_n(\xi u)$, and the output distribution of the modified algorithm should be exactly the same as that of the original algorithm. For the point in X, after the sieve, if $\mathbf{u}_i = \mathbf{u}_j$, we operate the modified second loop. With probability $1/2$, one of the point is changed to $\mathbf{u}_i \pm \mathbf{s}$ (or $\mathbf{u}_j \pm \mathbf{s}$) and the other remains unchanged. On this condition, computing $\mathbf{u}_i - \mathbf{u}_j$ for all points left we can get the shortest vector. Since the modified algorithm does not change the distribution of the output, we get the conclusion as we expected. Product the three probability together, the successful probability of algorithm 2 is $\frac{1}{8}$. □

Next we move to the proof of lemma 11. We need the following lemma in our proof.

Lemma 13 *[16] Let $E \subseteq \mathbb{R}^n \setminus \{\mathbf{0}\}$. If there exists $\phi_0 > 0$ such that for any $\mathbf{u}, \mathbf{v} \in E$, we have $\phi_{\mathbf{u},\mathbf{v}} > \phi_0$, then $|E| \leq 2^{cn+o(n)}$ with $c = -\frac{1}{2}\log_2[1 - \cos(min(\phi_0, 62.99^o))] - 0.099$.*

Proof of Lemma 11 To estimate the number of points in List T, we first bound the norm of points. After the sample algorithm, we have $\|\mathbf{t}' - \mathbf{t}\| \leq \xi\mu$, and the LLL reduced basis guarantee $\|\mathbf{t}'\| \leq 2^{O(n)}\mu$. According to the first loop of the algorithm, we have $c\mu \leq \|\mathbf{t}_i\| \leq (2^{O(n)} + \xi)\mu$. Divide this space into a polynomial number of spherical shells

$$T_d = \{\mathbf{t}_i \in T | d\mu < \|\mathbf{t}_i\| \leq (1 + \frac{1}{n})d\mu\}.$$

So we just need to consider the quantity of points in every spherical shell.

Since $\|\mathbf{t}_i - \mathbf{t}_j\| > c\mu$, $\langle \mathbf{t}_i, \mathbf{t}_j \rangle < \frac{\|\mathbf{t}_i\|^2 + \|\mathbf{t}_j\|^2 - c^2\mu^2}{2}$ holds. For a fixed spherical shell, denote $R = d\mu$, so $R < \|\mathbf{t}_i\|, \|\mathbf{t}_j\| \leq (1 + o(1))R$.

$$\cos(\phi_{\mathbf{t}_i,\mathbf{t}_j}) \leq \frac{\|\mathbf{t}_i\|^2 + \|\mathbf{t}_j\|^2 - c^2\mu^2}{2\|\mathbf{t}_i\|\|\mathbf{t}_j\|} \leq 1 - \frac{c^2\mu^2}{2R^2} + o(1).$$

On the other hand, the reduction condition gives

$$\langle \mathbf{t}'_i, \mathbf{t}_j \rangle < \frac{(1-\delta^2)\|\mathbf{t}'_i\|^2 + \|\mathbf{t}_j\|^2}{2},$$

therefore $\langle \mathbf{t}_i, \mathbf{t}_j \rangle < \frac{(1-\delta^2)\|\mathbf{t}'_i\|^2 + \|\mathbf{t}_j\|^2}{2} + \|\mathbf{t}_j\|\xi\mu$, so taking $\delta = 1 - \frac{1}{n}$, we have

$$\cos(\phi_{\mathbf{t}_i,\mathbf{t}_j}) \leq \frac{(1-\delta^2)\|\mathbf{t}'_i\|^2 + \|\mathbf{t}_j\|^2 + 2\|\mathbf{t}_j\|\xi\mu}{2\|\mathbf{t}_i\|\|\mathbf{t}_j\|} \leq \frac{1}{2} + \frac{\xi\mu}{R} + o(1).$$

Combining the two bounds on $\cos(\phi_{\mathbf{t}_i,\mathbf{t}_j})$, for all the spherical shells we get

$$\cos(\phi_{\mathbf{t}_i,\mathbf{t}_j}) \leq \max_R \min\{1 - \frac{c^2\mu^2}{2R^2}, \frac{1}{2} + \frac{\xi\mu}{R}\}.$$

The first term gets larger and the second term gets smaller as R increases. So the maximal value is obtained when $1 - \frac{c^2\mu^2}{2R^2} = \frac{1}{2} + \frac{\xi\mu}{R}$, $\frac{\mu}{R} = \frac{\sqrt{\xi^2+c^2}-\xi}{c^2}$. Using the lemma 13 we get the conclusion. $\square$