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 scheme related to $\mathcal{L}$. This 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. For some γ', 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 exploiting 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, as well as their approximate variants.

Successive minima in a lattice is a sequence $\{\lambda_i\}$ where λ_i is the radius of the smallest ball centered in the origin containing i linearly independent lattice vectors. In this paper, we emphasize that λ_i-gap (defined as λ_i/λ_1) provides extra 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 $\lambda_2/\lambda_1 > \gamma$, where $\gamma = n^c$, c is a positive constant). Lyubashevsky and Micciancio [25] introduced a reduction from $GapSVP_{2\gamma\sqrt{n/\log n}}$ (a decision variant of approximate SVP) to $uSVP_\gamma$, which confirms 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 [21]. It is obvious that the hardness of uSVP$_\gamma$ depends on the size of γ.

Some cryptographic schemes face serious security problems because of the large gap between λ_2 and λ_1 in their corresponding lattices(see [12]). For examples, general knapsack cryptosystems [27,7] are vulnerable to low density attacks because of the large λ_2-gap in the lattices on which they based; the large λ_2-gap in the embedding lattice of GGH [11,35] makes it easier to search the shortest vector. Even for the high density knapsack lattice with gap close to 1, there is a broadcast attack to enlarge the gap [40]. For the public-key cryptosystem NTRU, Coppersmith and Shamir [9]

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

constructed a cryptographic lattice with dimension $2N$ to analyze its security. A heuristic analysis reported in [17] indicates that NTRU lattice has λ_{N+1}-gap. Hence 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-based cryptanalysis.

For approximate SVP, the LLL basis reduction algorithm [23] achieves an exponential approximation factor in polynomial time, where the output vector $\mathbf{b}_1$ satisfies $\frac{\|\mathbf{b}_1\|}{\lambda_1} \leq \left((1+\varepsilon)\sqrt{\frac{4}{3}}\right)^{\frac{n-1}{2}}$ (ε is an arbitrary positive constant). Schnorr [44] generalized LLL algorithm to blockwise reduction. Assume that the block size is k, then there is a balance between the approximation factor $O((6k^2)^{\frac{n}{k}})$ and the running time that is determined by calling $poly(n)$ times k-dimensional exact SVP algorithm. In theory, the fastest known search for approximate SVP is the algorithm proposed by Gama and Nguyen [13]. It outputs a basis with the first vector $\mathbf{b}_1$ satisfying $\frac{\|\mathbf{b}_1\|}{\lambda_1} \leq ((1+\varepsilon)\gamma_k)^{\frac{n-k}{k-1}}$, at the cost of invoking $poly(n)$ times k-dimensional SVP-subroutine, where γ_n is Hermite's constant [10,30]. When $k = \alpha n$, the approximation factor decreases to $poly(n)$ with $2^{O(n)}$ operations. Recently, in [19], Hanrot, Pujol and Stehl*é* proved that BKZ with early termination can achieve the same time-quality trade-off as algorithm in [13]. Usually, basis reduction algorithms behave better than their proved worst-case theoretical bounds. In [12], Gama and Nguyen gave a significant assessment on the actual behavior of lattice reduction algorithms, based on extensive experiments. The best algorithm known in practice is the improved version of Schnorr-Euchner's BKZ [45] algorithm proposed by Chen and Nguyen [8] in 2011.

For exact SVP, there are many deterministic enumeration algorithms [20,38,45], with computational time ranging from $2^{O(n^2)}$ 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 reducing the time complexity to $2^{O(n)}$ with $2^{O(n)}$ space. Micciancio and Voulgaris [33] introduced another 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 [39] using the birthday attack. Also, under some random assumptions, there are heuristic versions of AKS sieve algorithm [36,47] and ListSieve [33], which are more efficient in practice. The best algorithm for most lattice problems, including SVP, SMP (Successive Minima Problem) and CVP was recently proposed by Micciancio and Voulgaris [34]. The algorithm is deterministic and based on Voronoi cell computation, whose running time is $\widetilde{O}(2^{2n+o(n)})$. [4] The survey [18] provides a detailed description on existing SVP algorithms.

This paper concerns short vectors for lattices with gaps. Firstly we compute a lower bound of the λ_2-gap for an embedding lattice from LWE problem [43]. For a set of parameters suggested in [15], the λ_2-gap of this lattice is larger than $7.3\log_2^2 m$, where m is the dimension of lattice. Furthermore, combining with the evaluation on the performance of lattice reduction [12], we estimate the range of the parameter related to the error vector that can be attacked by finding the shortest vector in the embedding lattice. This result provides a useful security assessment of the LWE-based schemes. Also, an improved reduction from BDD_{γ_1} to $uSVP_\gamma$ with larger λ_2-gap for some γ_1 is given.

Secondly, we show that gaps among successive minima lead to a more efficient SVP algorithm. Our purpose is to reduce the n-dimensional SVP to the SVP of sublattices with reduced-dimension, and then invoke the best deterministic SVP algorithm described in [34] to find the solution. The complexity is determined by the location of the gap and its size. For example, when $\lambda_{0.4n+1} >$

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

$c(n)\lambda_1$, and $c(n) = \log_2 n$, the time complexity bound of solving SVP decreases to $2^{0.802n+o(n)}$. This means that not only does the presence of a gap between λ_2 and λ_1 affect its security, the gaps between other successive minima also raise security problem. Although, it is believed there is no gap in random lattices, cryptographic lattices tend to possess this special property. To the best of our knowledge, the algorithm in this paper is the first SVP algorithm exploiting 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 problems. 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. Section 5 concludes this paper.

2 Notations and Background

Let $B = \{\mathbf{b_1}, \ldots, \mathbf{b_n}\} \subseteq \mathbb{R}^m$ consist of n linearly independent vectors. The 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 integer n and m are called its rank and dimension. If $m = n$, we say that the lattice is full-rank. Without loss of generality, we only consider the shortest vector problem in the full-rank lattices, since the other cases can be converted to a full-rank lattice with dimension n. The fundamental parallelepiped $\mathcal{P}(B)$ is defined to be $\{\Sigma_i x_i \mathbf{b}_i : 0 \le x_i < 1\}$ and $\mathcal{P}_{1/2}(B) = \{\Sigma_i x_i \mathbf{b}_i : -\frac{1}{2} \le x_i < \frac{1}{2}\}$. The volume of $\mathcal{P}(B)$ is called determinant of the lattice. The dual of a lattice $\mathcal{L}$ in $\mathbb{R}^m$, is defined to be $\mathcal{L}^* = \{\mathbf{x} \in \mathbb{R}^m : \forall \mathbf{v} \in \mathcal{L}, \langle \mathbf{x}, \mathbf{v} \rangle \in \mathbb{Z}\}$.

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., $\|\mathbf{x}\|_\infty = \max |x_i|$.
- $B_n(\mathbf{x}, r)$ denotes the n-ball centered at $\mathbf{x}$ with radius r, $\mathbf{x}$ is omitted when it is the origin.
- $N(n, \mathbf{x}, R^2)$ is the number of integer points in $B_n(\mathbf{x}, R)$, i.e.,

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

It is known that when $R > \sqrt{n/2}$, $N(n, \mathbf{x}, R^2) \le (\frac{2\pi e^{1+2w}}{n})^{\frac{n}{2}} R^n$, where w is a positive constant smaller than 1.024×10^{-4} [31].

Most of the worst-case to average-case reductions operate on q-ary lattices $\mathcal{L}$[1,43] which are defined as $q\mathbb{Z}^m \subseteq \mathcal{L} \subseteq \mathbb{Z}^m$. Given a matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, for some integers q, m, n, the following two types of m-dimensional q-ary lattices will be used in our discussion.

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

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

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

mod q. It is known that for random $\mathbf{A}$, we have $det(\Lambda_q(\mathbf{A}^T)) = q^{m-n}$ and $det(\Lambda_q^{\perp}(\mathbf{A}^T)) = q^n$, with high probability.

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 (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 non-zero $\mathbf{u} \in \mathcal{L}$.
- **Successive Minima Problem(SMP)**: Given a basis of a lattice $\mathcal{L}$, find n linear independent 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 a 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 a lattice vector closest to $\mathbf{t}$.

Given a basis of an n-dimensional lattice, the LLL algorithm [23] produces another basis of the lattice which consists of shorter vectors and is referred to as a LLL-reduced basis. Moreover,

Lemma 1. *[23] Let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ be an LLL-reduced basis, then we have $\|\mathbf{b}_j\| \leq \left((1+\varepsilon)\sqrt{\frac{4}{3}}\right)^{\frac{n-1}{2}} \lambda_j$.*

According to the above properties of LLL-reduced bases, for a fixed i $(1 \leq i \leq n)$, we can determine a μ_i such that $\lambda_i \leq \mu_i \leq (1+\frac{1}{n})\lambda_i$ by polynomial trials of μ_i. This approximation is essential in the reduction from 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 their best known theoretical upper bounds. In [12], based on abundant experiment, Gama and Nguyen pointed out the Hermite factor $\|\mathbf{b}_1\|/(det)^{\frac{1}{m}}$ is exponential δ^m in the dimension m, where $\mathbf{b}_1$ is the output of the algorithm and δ is called the root Hermite factor(δ varies in different algorithms, which is 1.0219 for LLL and 1.0128 for BKZ-20.) They also argued that the algorithm could achieve the shortest vector in the case of $\lambda_2/\lambda_1 > \tilde{\delta}^m$, for some $\tilde{\delta}$ slightly smaller than δ. Recently, in [8], Chen and Nguyen give a more detailed analysis of Hermite factor and running time of their basis reduction algorithm which is the best known algorithm in practice. In [32] and [28], the authors applied Hermite factor to evaluate the efficiency of their attacks on LWE. In Section 3.1, combining with the lower bound of the λ_2-gap in LWE embedding lattice, we will also use this Hermite factor to analyze the efficiency of solving LWE by embedding technique.

It is easy to compute a basis for the intersection of a lattice and a subspace, as indicated by the next lemma.

Lemma 2. *[29] There is a polynomial time algorithm, that inputs $\mathcal{L}$ with basis $\mathbf{B} \in \mathbb{Q}^{m \times n}$ and a linear subspace $S \subset \mathbb{Q}^m$, outputs a basis $\tilde{\mathbf{B}}$ for the sublattice $S \cap \mathcal{L}(\mathbf{B})$.*

Finally in this subsection, we shall describe the conceptual modification τ which was first presented by Regev [41]. Here, we introduce the version of Pujol and Stehlé [39]. Let $\mathbf{s}$ be a shortest vector and $I_{\mathbf{s}} = \{\mathbf{x} \in B_n(\xi\mu) : \mathbf{x} + \mathbf{s} \in B_n(\xi\mu)\}$, where μ is an approximate value of λ_1 and ξ is a fixed positive parameter. Let $\tau : B_n(\xi\mu) \longrightarrow B_n(\xi\mu)$ be such that $\tau(\mathbf{x}) = \mathbf{x} + \mathbf{s}$, if $x \in I_{\mathbf{s}}$; $\tau(\mathbf{x}) = -\mathbf{x}$, if $x \notin I_{\mathbf{s}}$. This transformation τ preserves the uniform distribution on $B_n(\xi\mu)$. We will use it to prove the success probabilities of our approximate SVP algorithm and reduced-dimension algorithm in the next section.

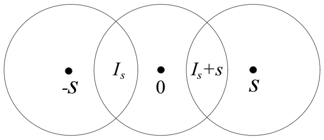

Fig. 1. τ Transformation

3 $\boldsymbol{\lambda_2}$-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 [35] and PJH [16] are also successfully attacked. The main reason is that the λ_2-gaps of the corresponding cryptographic lattices are large enough to leak substantial information. In this section, we will discuss the λ_2-gaps of the lattices applied to analyze LWE-based cryptosystem. Furthermore, we present a reduction from BDD_{γ_1} to $uSVP_{\gamma}$ with larger λ_2-gap for some γ_1.

3.1 $\boldsymbol{\lambda_2}$-Gap for LWE-based Lattice

The computational infeasibility of learning with errors (LWE) problem [43] guarantees the security of several cryptographic schemes [15,28,37,43]. The input to this problem is a pair $(\mathbf{A}, \mathbf{v} = \mathbf{A}\mathbf{s} + \mathbf{e})$, where $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ 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 is to recover $\mathbf{s}$. LWE problem can be regarded as a BDD instance in q-ary lattice $\Lambda_q(\mathbf{A}^T) = \{\mathbf{y} \in \mathbb{Z}^m : \mathbf{y} = \mathbf{A}\mathbf{s} \mod q \text{ for } \mathbf{s} \in \mathbb{Z}_q^n\}$. $\mathbf{v}$ is a target vector which is obtained by perturbing a randomly-chosen lattice vector $\mathbf{A}\mathbf{s} \in \Lambda_q(\mathbf{A}^T)$ with some small amount of noise $\mathbf{e}$. The perturbation is small enough that $\mathbf{A}\mathbf{s}$ is indeed the vector in $\Lambda_q(\mathbf{A}^T)$ closest to the perturbed point $\mathbf{v}$.

In this paper, we consider the error vector $\mathbf{e}$ samples from the discrete Gaussian distribution on $D_{\mathbb{Z}^m,s}$ (because s is relatively small compared to q, this distribution can be seen as on $\mathbb{Z}_q^m$), where for a real $s > 0$, n-dimensional lattice $\mathcal{L}$, the discrete Gaussian distribution $D_{\mathcal{L},s}$ is given by $D_{\mathcal{L},s}(\mathbf{x}) = \frac{\rho_{\mathbf{s}}(\mathbf{x})}{\rho_{\mathbf{s}}(\mathcal{L})}$ with $\mathbf{x} \in \mathcal{L}$, $\rho_{\mathbf{s}}(\mathbf{x}) = e^{-\pi\|\mathbf{x}/s\|^2}$ and $\rho_{\mathbf{s}}(\mathcal{L}) = \sum_{x\in\mathcal{L}} \rho_{\mathbf{s}}(\mathbf{x})$. In fact, our estimation of the λ_2-gap in LWE lattice can be applied to other error distributions provided that the length of the error has an appropriate bound.

The hardness of LWE was studied in [43] and it is proved that for the discrete Gaussian distribution $\chi = D_{\mathbb{Z}^m,\alpha q}$ with $\alpha q \geq 2\sqrt{n}$, the search-LWE is at least as hard as quantumly approximating a worst-case (the decision variant of) SVP or (the computational variant of) SIVP of n-dimensional lattice to within an approximation factor $\tilde{O}(n/\alpha)$. In [37], Peikert presented a classical reduction for (the decision variant of) SVP at the cost of increasing q. Recently, at STOC 2013, Brakerski et al gave a real classical reduction [6]. Cryptanalysts proposed several new methods to solve the LWE problem. Micciancio and Regev proposed to distinguish $\mathbf{v}$ from uniform distribution by finding a short vector in its dual lattice [32]; Lindner and Peikert applied a variant of NearestPlane algorithm on search-LWE [28]; Arora and Ge proposed a linearization technique [3] whose complexity depends on the length of the error .

Let us briefly describe the embedding technique used in [21] for reducing CVP to SVP. Given a lattice $\mathcal{L}$ with basis $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_m]$, and a target vector $\mathbf{t} \in \mathit{span}(B)$, the embedding method is to construct a new lattice $\mathcal{L}'$ with basis $\mathbf{B}' = [\mathbf{b}'_1, \mathbf{b}'_2, \cdots, \mathbf{b}'_{m+1}]$:

$$\begin{aligned} \mathbf{b'_1} &= (b_{11}, b_{12}, \ldots, b_{1m}, 0) \\ &\vdots \\ \mathbf{b'_m} &= (b_{m1}, b_{m2}, \ldots, b_{mm}, 0) \\ \mathbf{b'_{m+1}} &= (t_1, t_2, \ldots, t_m, \beta), \end{aligned} \tag{1}$$

where β is a parameter to be determined. If the distance between the target vector and the lattice is small enough, finding the shortest vector in this embedding lattice implies the solution to the CVP instance.

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

We need the following Lemma 3 and Lemma 4 to estimate the λ_2-gap for this lattice.

Lemma 3. *Let n, m be integers, and q be a prime such that $m > n$ and $q^{1-\frac{n+c}{m}} > \sqrt{\pi e^{1+2w}}$ for some positive constants c and w with $w < 1.024 \times 10^{-4}$. Let $\mathbf{A}$ be chosen uniformly from $\mathbb{Z}_q^{m\times n}$. Then for any $\mathbf{x} \in \mathbb{Z}^m$ we have, with probability bigger than $1 - q^{-c}$, $\min\limits_{a\in\varLambda_q(\mathbf{A}^T)\setminus\{\mathbf{x}\}} \|\mathbf{a} - \mathbf{x}\| \geq \min\{\sqrt{\frac{m}{2\pi e^{1+2w}}}q^{1-\frac{n+c}{m}}, q\}$. In particular, $\lambda_1(\varLambda_q(\mathbf{A}^T)) \geq \min\{\sqrt{\frac{m}{2\pi e^{1+2w}}}q^{1-\frac{n+c}{m}}, q\}$.*

Proof. The assumption $q^{1-\frac{n+c}{m}} > \sqrt{\pi e^{1+2w}}$ implies $\sqrt{\frac{m}{2\pi e^{1+2w}}}q^{1-\frac{n+c}{m}} > \sqrt{\frac{m}{2}}$. Let $R = \min\{\sqrt{\frac{m}{2\pi e^{1+2w}}}q^{1-\frac{n+c}{m}}, q\}$, then $N(m, \mathbf{x}, R^2) \leq N(m, \mathbf{x}, (\sqrt{\frac{m}{2\pi e^{1+2w}}}q^{1-\frac{n+c}{m}})^2) \leq q^{m-n-c}$. For any $\mathbf{x} \in \mathbb{Z}^m$, denote $V = \{\mathbf{a} \in \mathbb{Z}^m | \, \|\mathbf{a}+\mathbf{x}\| \leq R\}$. We have $|V| = N(m, \mathbf{x}, R^2)$. For uniformly random choice of $\mathbf{A}$ and any non-zero $\mathbf{s}$, since q is a prime, $Pr\{\mathbf{As} = \mathbf{a} \mod q\} = q^{-m}$.

The probability that there exists a vector in $\varLambda_q(A^T) \cap V$ is

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

Remark 1. References [15] and [46] give probabilistic lower bounds for the minimum length in this lattice in l_∞ norm and Euclidean norm respectively. In [46], only the case $m \geq 5n\log_2 q$ is considered, and the bound is $0.07\sqrt{m}q$ with probability $1-q^{-n}$. For the same parameters and probability, taking $c = n$ in our conclusion, we obtain a better bound $\sqrt{\frac{m}{2\pi e^{1+2w}}}q^{1-\frac{n+c}{m}} = \frac{2^{-0.4}}{\sqrt{2\pi e^{1+2w}}}\sqrt{m}q \approx 0.18\sqrt{m}q$.

The tail bound for discrete Gaussian distribution in the next lemma will be useful in estimating the norm of the error vector.

Lemma 4. *[5] Let $d > 1, s > 0$ and n be a positive integer. Let $\mathbf{x} \in \mathbb{Z}^n$ be randomly chosen according to $D_{\mathbb{Z}^n,s}$. Then* $\Pr[\|\mathbf{x}\| \geq d\frac{s\sqrt{n}}{\sqrt{2\pi}}] \leq \Big(d \cdot \exp(\frac{1-d^2}{2})\Big)^n$.

It can be seen that $d \cdot \exp(\frac{1-d^2}{2}) < 1$ as $d > 1$. In fact, the function $f(t) = t \cdot \exp(\frac{-t^2}{2})$ is strictly decreasing on $[1, \infty)$.

Now we are ready to measure the lower bound of λ_2-gap in LWE-based lattice.

Theorem 1. *Let n, m be integers, and q be a prime such that $m > n$ and $q^{1-\frac{n+c}{m}} > \sqrt{\pi e^{1+2w}}$, where $c > 1$ and w is a positive constant less than 1.024×10^{-4}. Let $\mathbf{e}$ be distributed according to $D_{\mathbb{Z}^m,\alpha q}$. Then for all but a fraction less than q^{-c+1} of the LWE-based lattice and any positive number ε, the gap between λ_1 and λ_2 satisfies: $\frac{\lambda_2}{\lambda_1} > \frac{R}{(1+\varepsilon)\alpha q\sqrt{\frac{m}{2\pi}}}$, where $R = \min\{\sqrt{\frac{m}{2\pi e^{1+2w}}}q^{1-\frac{n+c}{m}}, q\}$.*

Proof. Let $\mathcal{L} = \Lambda_q(\mathbf{A}^T)$. Using Lemma 4 by setting $d = 1 + \frac{\varepsilon}{2}$, we know that $\|\mathbf{e}\| \leq (1 + \frac{\varepsilon}{2})\alpha q\sqrt{\frac{m}{2\pi}}$ holds with overwhelming probability.

Let $\mathbf{v} = \mathbf{A}^T\mathbf{s}+\mathbf{e}$. We form $\mathcal{L}'$ by specifying $\mathbf{t} = (t_1, t_2, \ldots, t_m)$ to be $\mathbf{v}$, i.e.,$\mathbf{b}'_{m+1} = (\mathbf{v}, \beta)$. Denote $\mathbf{u} = \mathbf{A}^T\mathbf{s}$. Then the lattice vector $(\mathbf{v} - \mathbf{u}, \beta) \in \mathcal{L}'$ has norm $\sqrt{\|\mathbf{e}\|^2 + \beta^2}$. Choose $\beta = \frac{\varepsilon}{2}\alpha q\sqrt{\frac{m}{2\pi}}$, then we see that with overwhelming probability,

$$\|(\mathbf{v} - \mathbf{u}, \beta)\| = \sqrt{\|\mathbf{e}\|^2 + \beta^2} \leq \|\mathbf{e}\| + \beta \leq (1 + \varepsilon)\alpha q\sqrt{\frac{m}{2\pi}}.$$

We will prove that this vector is a unique shortest vector and the λ_2-gap is larger than $\frac{R}{(1+\varepsilon)\alpha q\sqrt{\frac{m}{2\pi}}}$.

Write $\mathbf{y} \in \mathcal{L}'$ as $\mathbf{y} = \sum_{i=1}^{m+1} x_i\mathbf{b}'_i$. Since we are working on q-ary lattices, we can partition the lattice $\mathcal{L}'$ into q sets $D_0, D_1, \ldots, D_{q-1}$ where $D_j = \{\mathbf{y} \in \mathcal{L}' | x_{m+1} = j\}$. Set $V_j = \{\mathbf{a} \in \mathbb{Z}^m | \, \|\mathbf{a}+j\mathbf{t}\| \leq R\}$, then from Lemma 3, $Pr(\mathcal{L} \cap V_j \neq \emptyset) \leq q^{-c}$.

Now for any $\mathbf{y} = \sum_{i=1}^{m+1} x_i\mathbf{b}'_i \in \mathcal{L}'$, $\mathbf{y}$ can be written as $(\mathbf{a} + j\mathbf{t}, j\beta)$ with $x_{m+1} = j$, $\mathbf{a} \in \Lambda_q(\mathbf{A}^T)$. Let $W_j = \{\mathbf{y} \in \mathcal{L}' | \mathbf{y} = (\mathbf{a} + j\mathbf{t}, j\beta), \mathbf{a} \in \Lambda_q(\mathbf{A}^T), \|\mathbf{y}\| \leq R\}$ and $W = \cup_{j=0}^{q-1} W_j$. Then from the previous discussion, we see that, $Pr(W_j \neq \emptyset) \leq q^{-c}$. This implies that $Pr(W \neq \emptyset) \leq qq^{-c} = q^{-c+1}$.

Therefore, with probability less than q^{-c+1}, there exists a vector $\mathbf{y} \in \mathcal{L}'$ with norm less than R. This shows that in $\mathcal{L}'$, the gap between λ_1 and λ_2 is larger than $\frac{R}{(1+\varepsilon)\alpha q\sqrt{\frac{m}{2\pi}}}$.

Next,we apply the lower bound in Theorem 1 to the LWE embedding lattice on which a LWE-based scheme proposed in [15] is based, and achieve a concrete lower bound. Then, combining the Hermite factor of basis reduction algorithm, we give the error range of LWE that is vulnerable to embedding method.

In [15], a cryptosystem based on the LWE problem was proposed. In their setting, the set of parameters are selected as $m = 6n \log_2 q$ where q is a prime with $q \in [\frac{n^2}{2}, n^2]$, the error distribution is $D_{\mathbb{Z}^m,\alpha q}$ for $\alpha = \frac{1}{\sqrt{m}\cdot\log_2^2 m}$. Using our Theorem 1, we are able to show that this specific LWE problem can be converted to the problem of finding the shortest vector in a lattice with large λ_2-gap. Similar to the discussion of SIS problem in [32], we should point out that the hardness of the LWE problem cannot be increased by enlarging the dimension m, because we can transfer the original LWE instance to the one with smaller m. More precisely, instead of considering $\Lambda_q(\mathbf{A}^T)$ we work on $\Lambda_q(\mathbf{A'}^T)$ where $\mathbf{A}'$ is obtained from $\mathbf{A}$ by removing some of its rows. We shall prove that the embedding lattice of $\Lambda_q(\mathbf{A'}^T)$ has larger λ_2-gap.

Corollary 1. *The above LWE problem can be solved by using a lower-dimensional lattice whose embedding lattice has λ_2-gap bigger than $7.3\log_2^2 m$ with overwhelming probability.*

Proof. Taking the first m' rows of $\mathbf{A}$, we get an $m' \times n$ matrix $\mathbf{A}'$. Denote by $\mathcal{L}'_{m'}$ the embedding lattice of $\Lambda(\mathbf{A}'^T)$ and let $G_{m'}$ be the λ_2-gap of $\mathcal{L}'_{m'}$. We will choose a suitable $m' \leq m$ to maximize $G_{m'}$.

Let $c = n$, then from Theorem 1 we know that if $\sqrt{\frac{m'}{2\pi e^{1+2w}}}q^{1-\frac{2n}{m'}} < q$, we have $G_{m'} > \frac{1}{(1+\varepsilon)e^{\frac{1}{2}+w}\alpha q^{\frac{2n}{m'}}}$, which increases with m'; if $\sqrt{\frac{m'}{2\pi e^{1+2w}}}q^{1-\frac{2n}{m'}} > q$, the lower bound of $G_{m'}$ is $\frac{\sqrt{2\pi}}{(1+\varepsilon)\alpha\sqrt{m'}}$ that decreases with m'. As a result, the optimal m' can be obtained by taking $\sqrt{\frac{m'}{2\pi e^{1+2w}}} = q^{\frac{2n}{m'}}$. For $m = 6n\log_2 q, q \in [\frac{n^2}{2}, n^2]$, let $m' = 0.7n\log_2 q = \frac{0.7m}{6}$, then we have $\sqrt{\frac{m'}{2\pi e^{1+2w}}}q^{1-\frac{2n}{m'}} > q$ as $n > 104$. With probability close to 1, we have $G_{m'} > \frac{\sqrt{2\pi}}{(1+\varepsilon)\alpha\sqrt{m'}} \approx 7.3\log_2^2 m$.

Now based on the result above, we compare the embedding method and the distinguishing attack on the LWE problem (the decision version) proposed by Micciancio and Regev [32]. Given $\mathbf{v} = \mathbf{A}\mathbf{s} + \mathbf{e}$ with $\mathbf{A} \in \mathbb{Z}_q^{m\times n}, \mathbf{s} \in \mathbb{Z}_q^n$ and $\mathbf{e} \in \mathbb{Z}_q^m$ obeys $D_{\mathbb{Z}^m,\alpha q}$, the idea is to distinguish $\mathbf{v}$ with ones from the uniform distribution. According to [32] and the Hermite factor obtained in [12], for the security parameter $\alpha < \sqrt{\frac{ln(1/\eta)}{\pi}}\delta^{-m'}q^{-\frac{n}{m'}}$, their attack can solve the decision version of LWE with probability higher than $\eta \in (0,1)$. Here m' is an integer between n and m which maximizes the range for the attack, in other words, optimizes the attack. If we take $\eta = 0.9$, the distinguished range of α in [32] is

$$\alpha < 0.18\delta^{-m'}q^{-\frac{n}{m'}}.$$

Actually [32] provided some more precise discussions on this by pointing out when $\sqrt{n\ln q/\ln\delta} < m$ (resp. $\sqrt{n\ln q/\ln\delta} \geq m$), $m' = \sqrt{n\ln q/\ln\delta}$ (resp. $m' = m$) is the number to minimize $\delta^{m'}q^{\frac{n}{m'}}$, and hence one gets the optimal attack. We also remark that for the concrete parameters in [28,32,42], $\sqrt{n\ln q/\ln\delta} > n$ holds.

Now we argue that using the result from Theorem 1, in high success probability cases, it is possible to relax the range of α by the embedding attack. As in the discussion of Corollary 1, we take the first m' rows of A as A' to use the embedding attack in $\Lambda_q(A'^T)$. Let $m' = \sqrt{n\ln q/\ln\delta} \leq m$, then $\sqrt{\frac{m'}{2\pi e^{1+2w}}}q^{1-\frac{n+c}{m'}} < q$, as long as $n\ln q > 400$ (According to the discussion in [12], the root Hermite factor $\delta < 1.005$ seems totally out of reach). Since $\sqrt{\frac{m'}{2\pi e^{1+2w}}}q^{1-\frac{n+c}{m'}}$ is an increasing function of m', when $\sqrt{n\ln q/\ln\delta} > m$, taking $m' = m$, we still have $\sqrt{\frac{m'}{2\pi e^{1+2w}}}q^{1-\frac{n+c}{m'}} < q$. Then $R = \sqrt{\frac{m'}{2\pi e^{1+2w}}}q^{1-\frac{n+c}{m'}}$ (in Theorem 1). Together with the conclusion in [12] (discussed earlier), Theorem 1 indicates that using the embedding lattice, we can solve the LWE problem when $\frac{\sqrt{\frac{m'}{2\pi e^{1+2w}}}q^{1-\frac{n+c}{m'}}}{(1+\varepsilon)\alpha q\sqrt{\frac{m'}{2\pi}}} > \tilde{\delta}^{m'}$ (more concisely $\alpha < \frac{1}{e^{\frac{1}{2}+w}(1+\varepsilon)}\tilde{\delta}^{-m'}q^{-\frac{n+c}{m'}}$). For example, taking $c = 2$ and $\varepsilon = 0.1$, with the success probability $1 - \frac{1}{q}(> 0.9)$ we can have

$$\alpha < 0.55\tilde{\delta}^{-m'}q^{-\frac{n+2}{m'}}.$$

That is, our approach can be used to attack instances with larger range of α.

Remark 2. In practice, the other algorithms for LWE are the NearestPlanes algorithm proposed by Lindner and Peikert [28] and the enumeration with pruning algorithm proposed by Liu and Nguyen [26]. The former one combines the NearestPlane algorithm and the enumeration algorithm to balance time/success tradeoff. The later one randomizes the Lindner and Peikert's algorithm and consider the natural adaptation of Gama-Nguyen-Regev's pruned enumeration [14] to BDD. In fact, this method can balance the time of enumeration and basis reduction flexibly. However it is hard to give the upper bounds of the attacking ranges of these two algorithms. In the previous discussion, what we have done is evaluating the λ_2-gap to analyze the efficiency of the attack using embedding lattice. In practice, our attack may not be better than some of the existing attacks, however we have established a theoretical upper bound in our setting.

3.2 Reduction from BDD to uSVP Revisited

Since LWE is a BDD instance, we estimate the hardness of solving LWE by computing 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, the larger the λ_2-gap is, the more easily the BDD instance can be solved. From the cryptanalysis point of view, given a fixed γ_1, the problem is easier if γ is larger. In other words, fixing γ, larger γ_1 means better reduction.

The following elegant reduction is presented in [25]:

Lemma 1 *[25] 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 to this reduction. We assume that the distance between $\mathbf{t}$ and $\mathcal{L}$ is u.

Given a basis of an n-dimensional lattice, the famous LLL algorithm [23] of Lenstra, Lenstra and Lováz computes an δLLL-reduced basis with $\delta \in (\frac{1}{4}, 1)$, then $\| \mathbf{b}_1 \| \leq (2/\sqrt{4\delta - 1})^{n-1}\lambda_1$. NearestPlane algorithm [4] with δLLL-reduced basis approximately solves CVP within a factor $2(2/\sqrt{3})^n$, for $\delta = 1/4 + (3/4)^{n/(n-1)}$. So we get d satisfying $\frac{d}{2(2/\sqrt{3})^n} < u \leq d$. By dividing the interval $(\frac{d}{2(2/\sqrt{3})^n}, d]$ into polynomially many parts $(\frac{d}{2(2/\sqrt{3})^n(1-\frac{1}{n})^i}, \frac{d}{2(2/\sqrt{3})^n(1-\frac{1}{n})^{i+1}}]$, guessing $u \in (\frac{d}{2(2/\sqrt{3})^n(1-\frac{1}{n})^{i_0}}, \frac{d}{2(2/\sqrt{3})^n(1-\frac{1}{n})^{i_0+1}}]$, and taking $u_0 = \frac{d}{2(2/\sqrt{3})^n(1-\frac{1}{n})^{i_0+1}}$, we have $u \leq u_0 < \frac{u}{1-\frac{1}{n}}$.

We construct the embedding lattice $\mathcal{L}'$ by specifying $\mathbf{b}'_{n+1} = (\mathbf{t}, ku_0)$ with k to be determined. The case of $k = 1$ corresponds to the reduction of $BDD_{\frac{1}{2\gamma}}$ to $uSVP_\gamma$ in [25]. By varying k, we can improve this reduction as long as $\gamma < 1.5321$.

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

Lemma 5. *[25] 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 as follows:

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

The proof is an adaptation of that of Theorem 1 in [25] (i.e.,Lemma 1) and it is given in the appendix.

It can be seen that the proof of Theorem 2 is valid for $1 < \gamma < 1.9318$. If $\gamma < 1.5321$, we have $\gamma_1 > \frac{1}{2\gamma}$. This improves the reduction of [25].

From the proof of Theorem 2, we can get the following corollary whose proof is provided in the appendix.

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

It is noted that the reduction of $\mathrm{BDD}_{\frac{1}{2\gamma}}$ to uSVP_γ in [25] 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 $\boldsymbol{\lambda_{\varepsilon n+1}}$-Gap

In this section, our purpose is to obtain a shortest vector of lattice with $\lambda_{\varepsilon n+1}$-gap by exploiting a modified sieve for approximate SVP and invoking the best known SVP algorithm [34] to find the shortest vector in reduced-dimensional sublattices. The presence of $\lambda_{\varepsilon n+1}$-gap makes it possible to sieve enough c-approximate shortest vectors which fall into a reduced-dimensional lattice.

4.1 The ListSieve-Birthday Algorithm

In this subsection we first recall the ListSieve-Birthday algorithm. We will present a modified version (the approximate SVP algorithm) in the next section. We also need some technical lemmas from [39] for proving the correctness of our approximate SVP algorithm.

We start with two routines, the Sample Algorithm and the Reduction Algorithm, as they are needed by the ListSieve-Birthday algorithm and our Algorithm 1.

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

```
Reduction Algorithm
Input: A pair $(\mathbf{u}, \mathbf{u}')$, a List $T \subseteq \mathcal{L}$ and reduced factor $\delta < 1$
Output: A reduced pair $(\mathbf{u}, \mathbf{u}')$
While $(\exists \mathbf{w} \in T) : \|\mathbf{u}' - \mathbf{w}\| \leq \delta \|\mathbf{u}'\|$
    $(\mathbf{u}, \mathbf{u}') \longleftarrow (\mathbf{u} - \mathbf{w}, \mathbf{u}' - \mathbf{w})$
end while
Return $(\mathbf{u}, \mathbf{u}')$
```

We remark that, in the reduction algorithm, the operations are determined by the perturbed vector $\mathbf{u}'$ instead of the lattice vector $\mathbf{u}$. If the norm of the perturbation $\mathbf{x}$ is large enough, a given perturbed vector can sometimes be obtained from several lattice vectors. This observation is crucial to our proof of correctness in the next section.

The ListSieve-Birthday Algorithm

```
Input: An LLL reduced basis $\mathbf{B}$, $N_1$,$N_2$,$\gamma > 1$, reduced factor $\delta < 1$, $\frac{\gamma}{2} > \xi > \frac{1}{2}$, $\mu \simeq \lambda_1$
Output: A shortest non-zero lattice vector
$T \longleftarrow \emptyset$,$U \longleftarrow \emptyset$
for $i = 1$ to $N_1$ do
    $(\mathbf{t}_i, \mathbf{t}'_i) \longleftarrow$ Reduction(Sample(B,$\xi\mu$),T,$\delta$)
    If $\|\mathbf{t}_i\| > \gamma\mu$ then
        $T \longleftarrow T \cup \{\mathbf{t}_i\}$
    end if
end for
for $i = 1$ to $N_2$ do
    $(\mathbf{u}_i, \mathbf{u}'_i) \longleftarrow$ Reduction(Sample(B,$\xi\mu$),T,$\delta$)
    $U \longleftarrow U \cup \{\mathbf{u}_i\}$
end for
find closest distinct points $(\mathbf{u}_i, \mathbf{u}_j)$ in $U$
Return $\mathbf{u}_i - \mathbf{u}_j$
```

It can be seen that the ListSieve-Birthday algorithm has two loops. The first loop constructs a list T by reducing each randomly generated vector with vectors previously added to the list. The second loop produces another list U whose elements are reduced in terms of the list T. This implies that the vectors in U are both short (with high probability) and independent and identically distributed.

In [39], it is proved that with suitable choices of the parameters N_1,N_2,γ, δ, ξ and μ, the ListSieve-Birthday algorithm can be used to solve SVP with probability $1 - 2^{-\Omega(n)}$[5] in time $2^{2.465n+o(n)}$. We will use the same set of parameters in our discussion in the next section. The precise choices of these parameters are given in the following lemmas.

Lemma 6. *[39] Let $c_l(\gamma, \xi) = -\frac{1}{2}\log_2(1 - \frac{2\xi}{\gamma}) + 0.401$. The List T in the algorithm contains at most $N_L(n) = 2^{c_l n+o(n)}$ vectors.*

Lemma 7. *[36] 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 $\mathcal{L}(\mathbf{B})$. Denote $I_{\mathbf{s}} = \{\mathbf{x} \in B_n(\xi\mu) : \mathbf{x} + \mathbf{s} \in B_n(\xi\mu)\}$. If $\mathbf{x}$ is chosen uniformly in $B_n(\xi\mu)$, then $Pr(\mathbf{x} \in I_{\mathbf{s}}) \geq \frac{1}{N_G}$, where $N_G = 2^{c_g n+o(n)}$.*

The parameter N_1 is related to the number $N_1^{max} = 4\lceil N_L N_G \rceil$. The following lemma is essentially the Lemma 6 and its remark of [39].

Lemma 8. *Let us consider the ListSieve-Birthday algorithm with N_1 chosen uniformly in the set $\{0, 1, 2, \ldots, N_1^{max} - 1\}$. Let E_i be the event $\|\mathbf{u}_i\| \leq \gamma\mu$, $i \leq N_2$, $p = Pr(E_i|\mathbf{x}_i \in I_{\mathbf{s}})$. Then with probability higher than $\frac{1}{2}$, $p > \frac{1}{2}$.*

4.2 The Approximate SVP Algorithm

Our Algorithm 1, which is used to obtain sufficiently many γ-approximate shortest vectors, is an SVP approximation algorithm modified from the ListSieve-Birthday algorithm [39]. Compared with the ListSieve-Birthday algorithm, Algorithm 1 terminates sieve process earlier and relaxes the birthday search. The number N_2 of sieved vectors is much smaller than that of the ListSieve-Birthday algorithm, which decreases the time complexity significantly.

[5] One writes $f(n) = \Omega(g(n))$, if there exist two positive constants c and n_0, for all $n \geq n_0$, $0 \leq cg(n) \leq f(n)$

Algorithm 1: The `Approximate SVP Algorithm`

```
Input: An LLL reduced basis $\mathbf{B}$, $N_1$,$N_2$,$\gamma > 1$, $d \geq 1$, dimension $n$, reduced factor $\delta < 1$, $\frac{\gamma}{1+1/\delta} > \xi > \frac{1}{2}$, $\mu \simeq \lambda_1$
Output: A shortest non-zero lattice vector or a pair of sets $(U, \overline{U})$ with $U$ the set of sieved lattice vectors and $\overline{U} = \{\mathbf{u} \in U : \|\mathbf{u}\| \leq \gamma\mu\}$
$T \longleftarrow \emptyset$,$U \longleftarrow \emptyset$,$\overline{U} \longleftarrow \emptyset$
for $i = 1$ to $N_1$ do
    $(\mathbf{t}_i, \mathbf{t}'_i) \longleftarrow$ Reduction(Sample(B,$\xi\mu$),T,$\delta$)
    If $\|\mathbf{t}_i\| > \gamma\mu$ then
        $T \longleftarrow T \cup \{\mathbf{t}_i\}$
    end if
end for
for $i = 1$ to $N_2$ do
    $(\mathbf{u}_i, \mathbf{u}'_i) \longleftarrow$ Reduction(Sample(B,$\xi\mu$),T,$\delta$)
    $U \longleftarrow U \cup \{\mathbf{u}_i\}$
    if $\|\mathbf{u}_i\| \leq \gamma\mu$ then
        $\overline{U} \longleftarrow \overline{U} \cup \{\mathbf{u}_i\}$
    end if
end for
find closest distinct points $(\mathbf{u}_i, \mathbf{u}_j)$ in $U$
if $\|\mathbf{u}_i - \mathbf{u}_j\| \leq \mu$
    Return $\mathbf{u}_i - \mathbf{u}_j$
else
    Return $(U$,$\overline{U})$
```

The sieve steps in our approximate SVP algorithm and the ListSieve-Birthday algorithm are the same except that we store an additional set $\overline{U}$ which is a subset of U satisfying $\overline{U} = \{\mathbf{u} \in U : \|\mathbf{u}\| \leq \gamma\mu\}$. So, the early analysis from Lemma 6, Lemma 7 and Lemma 8 applies to our Algorithm 1 as well.

In the description of our algorithm, we use the same notations as in Lemma 6, Lemma 7 and Lemma 8. The algorithm succeeds if a shortest non-zero lattice vector is returned or $\overline{U}$ contains at least d distinct lattice vectors whose perturbation $\mathbf{x}$ is in $I_{\mathbf{s}}$. Some parameters are given as follows: N_1 is chosen uniformly from the set $\{0, 1, 2, \ldots, N_1^{max} - 1\}$, $N_2 = 8dN_G$, $\delta = 1 - \frac{1}{n}$. Other parameters will be determined later.

The following lemma, which proves the correctness of the Approximate SVP Algorithm, is also needed by the main technical lemma (Lemma 11). Its proof is similar to that of Lemma 7 in [39].

Lemma 9. *Let $N_2 = 8dN_G$, and assume that n is sufficiently large. Then with probability higher than $\frac{1}{8}$, the algorithm succeeds.*

Proof. Let $\mathbf{s}$ be a shortest vector of the lattice whose norm is approximately μ, and define $I_{\mathbf{s}} = \{\mathbf{x} \in B_n(\xi u), \mathbf{x} + \mathbf{s} \in B_n(\xi u)\}$. According to Lemma 8, with probability higher than $\frac{1}{2}$, $Pr(\|\mathbf{u}_i\| \leq \gamma\mu | \mathbf{x}_i \in I_{\mathbf{s}}) \geq \frac{1}{2}$ holds. Since the vectors in the list U are independent and identically distributed and the relation $\mathbf{u}_i - \mathbf{u}'_i = \mathbf{x}_i$ is preserved during the sieve process, we see that

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

Let $X = \{i \leq N_2 : \|\mathbf{u}_i\| \leq \gamma\mu, \mathbf{x}_i \in I_{\mathbf{s}}\}$. Based on the analysis above, the random variable $|X|$ obeys a binomial distribution of parameter $p \geq \frac{1}{2N_G}$. Since the expectation and variance are

$\mathbb{E}(|X|) = pN_2$ and $\mathbb{D}(|X|) = p(1-p)N_2$ respectively, by Chebyshev's inequality we have

$$Pr(|X| \leq d) \leq Pr(||X| - \mathbb{E}(|X|)| \geq \mathbb{E}(|X|) - d) \leq \frac{\mathbb{D}(|X|)}{(\mathbb{E}(|X|) - d)^2} \leq \frac{4}{9d} \leq \frac{1}{2}$$

when $d \geq 1$.

That means we have $|X| > d$ with probability higher than $\frac{1}{2}$. The following discussion will be divided into two cases.

Case 1. If there are distinct $i, j \in X$ such that $\mathbf{u}_i = \mathbf{u}_j$, we claim that a shortest vector can be found by pairwise subtracting the elements in U with high probability.

We modify the Sample Algorithm in the second loop by applying τ with probability 1/2 on every perturbation $\mathbf{x}$. τ maintains the uniform distribution on $B_n(\xi\mu)$, and the output distribution of the modified algorithm should be exactly the same as that of the original algorithm. Furthermore, we have

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

This means that for $\mathbf{x} \in I_\mathbf{s}$, if the original Sample Algorithm returns $(\mathbf{u}, \mathbf{u}')$, then its modification (i.e., after τ transformation) outputs $(\mathbf{u}+\mathbf{s}, \mathbf{u}')$. Since in the Reduction Algorithm, the sieve makes its decision based on $\mathbf{u}'$ instead of $\mathbf{u}$, the τ transformation has no effect on the Reduction Algorithm.

Since $\mathbf{u}_i = \mathbf{u}_j$, with probability 1/2, $\mathbf{u}_i$ is changed to $\mathbf{u}_i + \mathbf{s}$, or $\mathbf{u}_j$ is changed to $\mathbf{u}_j + \mathbf{s}$, but not both, after using τ to the second loop. This means that the shortest vector $\mathbf{s}$ is in $\{\mathbf{w}_1 - \mathbf{w}_2 : \mathbf{w}_1, \mathbf{w}_2 \in U\}$. Since the modified algorithm does not change the the distribution in U, Algorithm 1 returns the shortest vector in this step as well.

Case 2. If for all distinct $i, j \in X$, $\mathbf{u}_i \neq \mathbf{u}_j$, then at least d distinct vectors whose perturbation $\mathbf{x}$ is in $I_\mathbf{s}$ are in $\overline{U}$.

Multiplying the three probabilities together, the success probability of Algorithm 1 is higher than $\frac{1}{8}$. □

Now, we are able to estimate the complexity of Algorithm 1.

Theorem 3. *Let* $c_{time} = \max\{2c_l(\gamma, \xi) + c_g(\xi), 2c_g(\xi)\}$ *and* $c_{space} = \max\{c_l(\gamma, \xi), c_g(\xi)\}$. *Let* $d = 2^{o(c_g n)}$. *Then with probability* $1 - 2^{-\Omega(n)}$*, Algorithm 1 succeeds with time* $2^{c_{time} n + o(n)}$ *and space* $2^{c_{space} n + o(n)}$.

Proof. The time complexity of the first loop in steps 2-7 is $N_1 N_L$, and that of the second loop of steps 8-14 is $N_2 N_L$. The complexity of steps 15-19 is N_2^2. So the total time complexity is $2^{c_{time} n + o(n)}$ as $c_{time} = \max\{2c_l(\gamma, \xi) + c_g(\xi), 2c_g(\xi)\}$. It is obvious that space complexity is $|T| + |U| = 2^{c_{space} n + o(n)}$, as $c_{space} = max\{c_l(\gamma, \xi), c_g(\xi)\}$.

Calling the algorithm n times ensures that it succeeds with probability exponentially close to 1. □

In the Table below, the column labeled "Algorithm 1" lists some optimal complexity bounds given in Theorem 4.5 for several choices of c and the corresponding ξ. We see that the complexity bound decreases obviously with the increase of c. In particular, if $c(n) = \Omega(\log_2 n)^6$, we can find the approximate shortest vector in time $\tilde{O}(2^{0.802n+o(n)})$ and space $\tilde{O}(2^{0.401n+o(n)})$.

[6] One writes $f(n) = \Omega(g(n))$ if there exist two positive constants c and n_0, for all $n \geq n_0$, $0 \leq cg(n) \leq f(n)$

If we are only interested in solving approximate SVP, then it is sufficient to find one non-zero vector that is shorter than $c\mu$. We modify Algorithm 1 by adding lattice vectors reduction before appending a point to the List which guarantees the distances between the points in the List are larger than $c\mu$. This property decreases the upper bound of the List size and reduces the time complexity accordingly. More precisely, 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, hence $\|\mathbf{t}'_i - \mathbf{t}_j\| > (1 - \frac{1}{n})\|\mathbf{t}'_i\|$ holds (taking $\delta = 1 - \frac{1}{n}$). Suppose $\mathbf{t}_i - \mathbf{t}_j = 0$, when n is large, we have

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

This is a contradiction. This implies that when the algorithm terminates at this step, a short non-zero vector whose length is less than $c\mu$ is obtained. When the first loop ends, if the algorithm does not 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 [33], the following lemma (the proof is given in the appendix) tells us that we can use a smaller c_l to bound the size of List T.

Lemma 10. *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 at most double the upper bound of time complexity. Other parts of the estimation of complexity bound are the same as Lemmas 7, 8, 9 and Theorem 3. For an approximation factor c, we list some optimal time complexity bounds in the column marked "Modified Algorithm" in the Table below. This algorithm also has time complexity bound $\tilde{O}(2^{0.802n+o(n)})$ and space complexity bound $\tilde{O}(2^{0.401n+o(n)})$, when $c(n) = \Omega(\log_2 n)$.

	Algorithm 1			Modified Algorithm		
c	ξ	time	space	ξ	time	space
2.71	0.6971	2.3655	0.9222	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

In our main algorithm in the next subsection, the improvement above helps to deal with 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.

Note that in [13], the best known approximation algorithm achieves $\frac{\|\mathbf{b}_1\|}{\lambda_1} \approx k^{\frac{n-k}{k-1}}$, with time complexity $\tilde{O}(2^{2k})$. So, when $c < n^{\gamma}$ with $\gamma \approx 1.5$, our algorithm performs better.

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

Assume $\varepsilon < 1$, and the $\lambda_{\varepsilon n+1}$-gap is bigger than γ, i.e., $\lambda_{\varepsilon n+1} > \gamma\lambda_1$. In this subsection, we shall use Algorithm 1 from the previous subsection to sieve enough lattice points that are in

$B_n(\gamma\lambda_1)$. The assumption on the gap indicates that these points are actually in the sublattice $span\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{\varepsilon n}\} \cap \mathcal{L}(\mathbf{B})$ of dimension εn, where $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n$ are linearly independent lattice vectors and $\|\mathbf{s}_i\| = \lambda_i$. As a result, solving the SVP of the original $\mathcal{L}$ can be reduced to SVP of the reduced-dimensional sublattices.

Algorithm 2 is our reduced-dimension algorithm for searching shortest lattice vector in the presence of $\lambda_{\varepsilon n+1}$-gap. This algorithm consists of three parts. The first part sieves enough short vectors which fall into a subspace. The core of the second part is to apply Lemma 2 (we call the corresponding algorithm for generating a basis for the sublattice the SubLattice Algorithm). Given the inputs of the lattice basis $\mathbf{B}$, a set of lattice vectors ($\overline{U}$ or $\tilde{U}$), the SubLattice Algorithm outputs a basis of a sublattice that may contain a shortest vector of the original lattice. The final part finds the shortest lattice vector in the reduced-dimensional sublattice by using the Deterministic SVP Algorithm of [34].

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

```
Input: A basis of n-dimensional lattice $\mathbf{B}$, $N_1$,$N_2$,$\gamma > 1$, $\varepsilon < 1$, reduced factor $\delta < 1$, $\frac{\gamma}{1+1/\delta} > \xi > \frac{1}{2}$, $\mu \simeq \lambda_1$
Output: A shortest non-zero lattice vector
1: $U \longleftarrow \emptyset$, $\overline{U} \longleftarrow \emptyset$
2: If Approximate SVP Algorithm($\mathbf{B}, N_1, N_2, \gamma, \varepsilon n^3, n, \delta, \xi, \mu$) returns a shortest non-zero
      lattice vector $\mathbf{s}$
3.   return $\mathbf{s}$
4. else
5.   $(U, \overline{U}) \longleftarrow$ Approximate SVP Algorithm($\mathbf{B}, N_1, N_2, \gamma, \varepsilon n^3, n, \delta, \xi, \mu$)
6:   $\tilde{\mathbf{B}} \longleftarrow$ SubLattice Algorithm($\mathbf{B}, span(\overline{U})$)
7:   if $dim(\tilde{\mathbf{B}}) = \varepsilon n$
8:     $\mathbf{s} \longleftarrow$ Deterministic SVP Algorithm($\tilde{\mathbf{B}}, \varepsilon n$)
9:   else
10:      randomly choose $\mathbf{v} \in \overline{U}$
11:      $\mathbf{s} \longleftarrow \mathbf{v}$
12:      for every $\mathbf{v} \in U \setminus \overline{U}$
13:        $\tilde{U} \longleftarrow \overline{U} \cup \{\mathbf{v}\}$
14:        $\tilde{\mathbf{B}} \longleftarrow$ SubLattice Algorithm($\mathbf{B}, span(\tilde{U})$)
15:        $\tilde{\mathbf{s}} \longleftarrow$ Deterministic SVP Algorithm($\tilde{\mathbf{B}}, \varepsilon n$)
16:        if $\| \tilde{\mathbf{s}} \| < \| \mathbf{s} \|$
17:          $\mathbf{s} = \tilde{\mathbf{s}}$
18:        end if
19:      end for
20:   end if
21:end if
22:return $\mathbf{s}$
```

If the Approximate SVP Algorithm returns a shortest lattice vector in step 3, the algorithm succeeds. In the following discussion in this subsetion, without loss of generality, we assume this case does not happen.

In the algorithm, we only need to construct at most N_2 sublattices with dimensions no more than εn. The next lemma shows that if d is taken to be εn^3, the shortest lattice vector must be in one of these sublattices.

Lemma 11. *There is a subset* $\{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_{\varepsilon n^3}\} \subset \overline{U} \setminus \{\mathbf{0}\}$*, with probability close to 1, the shortest vector* $\mathbf{s}$ *belongs to at least one of the sublattices of the following form: span*$(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{\varepsilon n^3}, \mathbf{v}) \cap \mathcal{L}(B)$*, where* $\mathbf{v} \in (U \setminus \overline{U}) \cup \{\mathbf{0}\}$*.*

Proof. From the proof of Lemma 9, we know that, in $\overline{U}$, there are at least $d = \varepsilon n^3$ different lattice vectors $\mathbf{u}_i$ satisfying $\mathbf{u}'_i - \mathbf{u}_i \in I_{\mathbf{s}}$. We will let R be such set, i.e.,

$$R = \{\mathbf{u} : \mathbf{u} \in \overline{U}, \mathbf{u}' - \mathbf{u} \in I_{\mathbf{s}}\}.$$

By taking the first d elements if necessary, we may assume that $|R| = d$. i.e., we can write

$$R = \{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_d\}.$$

Now let

$$S = \{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_M\}$$

be the set of all lattice vectors whose length is smaller than or equal to $\gamma\mu$. The set R produced by Algorithm 1 is thus a subset of S with d elements.

Due to the random nature of Algorithm 1, the set R obtained is also random. We assume the distribution for the i-th element $\mathbf{u}_i$ is $\mathcal{D}$ (we omit the subindex as it is observed that $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_d$ are independent and identically distributed).

Let H be the linear subspace generated by R, i.e., $H = span(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{\varepsilon n^3})$. Since $\lambda_{\varepsilon n+1} > \gamma\lambda_1$ and $\|\mathbf{u}_i\| \le \gamma\lambda_1$, we have $H \subseteq span\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{\varepsilon n}\}$, where $\mathbf{s}_i$ $(i = 1, \ldots, \varepsilon n)$ are linearly independent and $\|\mathbf{s}_i\| = \lambda_i$. This yields $dim(H) \le \varepsilon n$. We shall prove the lemma by considering an ascending chain of εn subspaces $H_i = span(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{in^2})$, $i = 1, 2, \ldots, \varepsilon n$ and $H_0 = span(\mathbf{0})$, similar to that in the proof of Corollary 3.16 of [43].

If for some $j_0 < \varepsilon n$, we have $\dim(H_{j_0}) = \varepsilon n$, then $\dim(H_{\varepsilon n}) = \varepsilon n$. This forces that the shortest lattice vector $\mathbf{s}_1 \in H$, as desired. Otherwise for all $j < \varepsilon n$, $\dim(H_j) < \varepsilon n$. We need to deal with two cases.

Case 1: The probability of $\mathbf{u} \in H_{\varepsilon n-1}$, where $\mathbf{u}$ obeys $\mathcal{D}$, is at most $1 - \frac{1}{n}$. In this case, we shall prove that with high probability, $\dim(H_{\varepsilon n}) = \varepsilon n$.

In fact, in this case we have that the probability of $\mathbf{u} \in H_j$, where $\mathbf{u}$ obeys $\mathcal{D}$, is at most $1 - \frac{1}{n}$, for every $j < \varepsilon n$. This implies that the probability that all $\mathbf{u}_{jn^2+1}, \mathbf{u}_{jn^2+2}, \cdots, \mathbf{u}_{jn^2+n^2} \in H_j (j = 0, 1, ..., \varepsilon n - 1)$ is less than $(1 - \frac{1}{n})^{n^2}$. So there is an i_0 with $jn^2 + 1 \le i_0 \le jn^2 + n^2$ such that $\mathbf{u}_{i_0} \notin H_j$, with probability higher than $1 - (1 - \frac{1}{n})^{n^2}$. In particular, we have $\dim(H_{j+1}) > \dim(H_j)$. Since $\dim(H_1) > \dim(H_0) = 0$, this means that $\dim(H_{\varepsilon n}) = \varepsilon n$ with probability higher than $\left(1 - (1 - \frac{1}{n})^{n^2}\right)^{\varepsilon n} \ge 1 - \frac{\varepsilon n}{e^n}$.

Case 2: The probability of $\mathbf{u} \in H_{\varepsilon n-1}$, where $\mathbf{u}$ obeys $\mathcal{D}$, is bigger than $1 - \frac{1}{n}$. In this case, we shall prove that with high probability, the shortest lattice vector $\mathbf{s} \in span(H \cup \{\mathbf{v}\})$ for some $\mathbf{v} \in (U \setminus \overline{U}) \cup \{\mathbf{0}\}$.

First, it is easy to see that the probability of $\mathbf{u} \in H$, where $\mathbf{u}$ obeys $\mathcal{D}$, is larger than $1 - \frac{1}{n}$. We will proceed our proof by showing that with overwhelming probability, there exist a $\mathbf{v} \in U$ and a subspace $H_0 \subset H$ such that $\mathbf{v} \in H_0 + \mathbf{s}$. Similar to the proof of Lemma 9, we will use the τ transformation to achieve this conclusion.

Once again, the Sample Algorithm is modified as follows: after choosing $\mathbf{x}$ uniformly from $B_n(\xi\mu)$, apply τ with probability 1/2 on every perturbation $\mathbf{x}$. Note that

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

This means that for $\mathbf{x} \in I_{\mathbf{s}}$, if the original Sample Algorithm returns $(\mathbf{w}, \mathbf{w}')$, the output of the its modified version is $(\mathbf{w}+\mathbf{s}, \mathbf{w}')$ after this transformation. We then modify Algorithm 1 by using this modified Sample Algorithm in the second loop. Because τ maintains the uniform distribution on $B_n(\xi\mu)$, the output distribution of the modified Algorithm 1 should be exactly the same as that of the original algorithm. We note that τ transformation does not affect the Reduction Algorithm, as we have argued in the proof of Lemma 9.

Running the modified Algorithm 1, we obtain the output $(U', \overline{U}')$ with $\overline{U}'$ containing sufficiently many vectors that are shorter than $\gamma\mu$. In a similar manner, we get a set R' that contains $d = \varepsilon n^3$ vectors, say

$$R' = \{\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2, \ldots, \tilde{\mathbf{u}}_{\varepsilon n^3}\}.$$

Let $H' = span(R')$ and consider $H_0 = H \cap H'$. Now let us prove that the probability of $\mathbf{u} \in H_0$, where $\mathbf{u}$ obeys $\mathcal{D}$, is bigger than $1-\frac{2}{n}$. In fact, for $\mathbf{u}$ being chosen from $\mathcal{D}$, similar to the situation for H, the probability of $\mathbf{u} \in H'$ is bigger than $1-\frac{1}{n}$. This means that the probability of $\mathbf{u} \in H^c \cup H'^c$ is smaller than $\frac{2}{n}$. The result follows immediately. Observe that the probability that at least one vector from R is in H_0 is $1-\left(\frac{2}{n}\right)^{\varepsilon n^3}$. So with probability higher than $\frac{1}{2}\left(1-\left(\frac{2}{n}\right)^{\varepsilon n^3}\right)$, there exists at least one vector $\mathbf{u}_{i_0} \in H_0 \cap \overline{U}$ which is changed to $\mathbf{v} = \mathbf{u}_{i_0} + \mathbf{s}$ after the τ transformation. This implies that the set U' contains the vector $\mathbf{v} \in H_0 + \mathbf{s}$. If $\mathbf{v} \in \overline{U}'$, the shortest vector $\mathbf{s}$ belongs to $span(\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2, \ldots, \tilde{\mathbf{u}}_{\varepsilon n^3}) \cap \mathcal{L}(B)$. Otherwise, $\mathbf{s} \in span(\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2, \ldots, \tilde{\mathbf{u}}_{\varepsilon n^3}, \mathbf{v}) \cap \mathcal{L}(B)$ and $\mathbf{v} \in (U \setminus \overline{U})$.

The proof is completed. □

To state the complexities of Algorithm 2, let us first fix some parameters used in the following theorem: let $c_{time}(\gamma, \xi, \varepsilon) = \max\{2c_l(c, \xi)+c_g(\xi), 2c_g(\xi), 2\varepsilon n+c_g n\}$, $c_{space}(\gamma, \xi, \varepsilon) = \max\{c_l(\gamma, \xi), c_g(\xi), \varepsilon n\}$. Now we are ready to prove:

Theorem 4. *When $\lambda_{\varepsilon n+1} > \gamma\lambda_1$, the time complexity of Algorithm 2 is $2^{c_{time}n+o(n)}$, and the space complexity is $2^{c_{space}n+o(n)}$.*

Proof. We have proved in Theorem 3 that the time complexity bound of steps 2 to 5 of Reduced-Dimension Algorithm is $2^{c_{time_1}n+o(n)}$, where $c_{time_1}(\gamma, \xi) = max\{2c_l(\gamma, \xi) + c_g(\xi), 2c_g(\xi)\}$. With $N_2 = 8d2^{c_g n}$, the time spent on step 6 is $2^{c_g n+o(n)}$. For steps 7 to 22, according to Lemma 11, the algorithm invokes at most N_2 times εn-dimensional exact SVP, which need $2^{2\varepsilon n+c_g n+o(n)}$ operations. Summing them up, we get the time complexity.

For space complexity, for steps 2 to 5, $2^{c_{space_1}n+o(n)}$ is needed, where $c_{space_1}(\gamma, \xi) = max\{c_l(\gamma, \xi), c_g(\xi)\}$. Steps 6 to 22 cost space $max\{2^{c_g n+o(n)}, 2^{\varepsilon n+o(n)}\}$. □

Remark 3. If we solve the approximate SVP by the AKS sieve (the approximate version of AKS is provided by [36]) instead of the ListSieve, $U = \overline{U}$ always holds. So in this case, we only need to consider one reduced-dimensional sublattice. But the time complexity bound of the AKS for approximate SVP is higher than that of the ListSieve. This difference becomes smaller with the increase of the approximation factor γ. On the other hand, c_g will decrease with the increase of γ, since the optimal ξ is also increasing with γ. Moreover, in most cases, getting these approximate short vectors costs more than finding the shortest vector in reduced-dimensional lattice. That is why we choose the ListSieve. The overall complexity bound of the algorithm is determined by the location as well as the size, of the gap.

4.4 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 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 4.3. To make the description of the recursive algorithm more clear, we regard the Approximate SVP Algorithm here as the approximate version of AKS which implies that we only need to consider one sublattice. To treat the general case, we can set $\varepsilon_{t+1} = 1$ and $\tilde{\mathbf{B}}_0 = \mathbf{B}$ in Algorithm 3.

```
Algorithm 3: Multi-Gap lattice Algorithm
Input: A basis of n-dimensional 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
$U \longleftarrow \emptyset$
    for $i = 0$ to $i = t - 1$
        $U_i \longleftarrow$ Approximate SVP Algorithm($\tilde{\mathbf{B}}_\mathbf{i}, N_1^i, N_2^i, c_{t-i}, \varepsilon_{t-i}n^3, \varepsilon_{t-i+1}n, \delta, \xi^{(i)}, \mu$)
        $\tilde{\mathbf{B}}_{i+1} \longleftarrow$SubLattice Algorithm($\tilde{\mathbf{B}}_\mathbf{i}, U_i, \varepsilon_{t-i}n$)
    end for
$\mathbf{s} \longleftarrow$ Deterministic SVP Algorithm($\tilde{\mathbf{B}}_\mathbf{t}, \varepsilon_1 n$)
```

Denote the time complexity of Approximate SVP algorithm for n-dimensional lattice as $\tilde{O}(2^{c'_{time1}(c,\xi)n+o(n)})$ and the space as $\tilde{O}(2^{c'_{space1}(c,\xi)n+o(n)})$. The total time and space complexity are $\tilde{O}(2^{c_{time}+o(n)})$ 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 based on Voronoi cell computation is $2^{2n+o(n)}$, the key parameter to compare our algorithm and the previous algorithm is γ_0 which is the minimal γ satisfying $c_{time1}(\gamma, \xi) < 2$. From Table 1 in Section 3.1, we see that γ_0 is 3.61 (for approximate AKS sieve, γ_0 has to be 3.97). For a lattice with $\lambda_i < \gamma_0\lambda_1$, for all $i \leq n - o(n)$, the Deterministic SVP Algorithm performs better. For this kind of lattices, the gaps are too small to be used to simplify the general algorithm for SVP.

5 Conclusion

In this paper, we first estimate the size of the λ_2-gap in the embedding lattice obtained from LWE problem. Solving SVP in this lattice would find the solution of corresponding LWE problem. Our analysis reveals that the error range of LWE attacked by embedding method is larger than that of the existing distinguishing attack in high success probability case .For general BDD problem, for some value of γ_1, we present an improved reduction from BDD_{γ_1} to $uSVP_\gamma$ by constructing an embedding lattice with larger λ_2-gap.

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

Although our algorithm decreases the upper bound of time complexity for SVP of lattice possessing gaps, as other random sieves, perturbation technique is used to prove the success probability which is the bottleneck of the algorithm and makes this algorithm impractical. But we believe this algorithm will motivate further research on finding practical algorithms for specific lattices with gaps.

6 Acknowledgments

We thank the anonymous referees for their careful reading and thoughtful and constructive comments.

References

1. M.Ajtai and C.Dwork, A public-key cryptosystem with worst-case/average-case equivalence. In STOC 1997. pp.284-293. El Paso, Texas, USA, 1997.
2. M.Ajtai, R.Kumar and D.Sivakumar, A sieve algorithm for the shortest lattice vector problem. In: STOC 2001. pp. 266-275. Heraklion, Crete, Greece, ACM, New York, 2001.
3. S.Arora and R.Ge. New algorithm for lerning in presence of errors. In ICALP 2011. LNCS, vol.6755, pp.403-415. Zürich, Switzerland 2011.
4. L.Babai, On lovasz' lattice reduction and the nearest lattice point problem. Combinatorica 6(1)(1986), pp.1-13.
5. W.Banaszczyk, New bounds in some transference theorems in the geometry of numbers. Mathematische Annalen, 296(4)(1993), pp.625-635.
6. Brakerski Z, Langlois A, Peikert C, et al. Classical hardness of learning with errors. In: Boneh D, Roughgarden T, Feigenbaum J, eds. STOC, ACM, 2013. 575–584.
7. M.J.Coster, A.Joux, B.A.La Macchia, A.M.Odlyzko, C.P.Schnorr, J.Stern, An improved low-density subset sum algorithm, Computational Complexity.2(1992), pp.97-186.
8. Y.M.Chen, P.Q.Nguyen, BKZ 2.0: Better Lattice Security Estimates. In ASIACRYPT 2011. pp.1-20. Seoul, Korea, 2011.
9. D.Coppersmith, A.Shamir, Lattice attacks on NTRU. In: EUROCRYPT 1997. LNCS, vol.1233, pp.52-61. Konstanz, Germany, 1997.
10. J.Conway, N. Sloane, Sphere Packing, Lattices and Groups. Springer-Verlag. Third edition(1998).
11. Goldreich O, Goldwasser S, Halevi S. Public-key cryptosystems from lattice reduction problems. In: Kaliski B S, (eds.). Proceedings of Advances in Cryptology-CRYPTO 1997, volume 1294 of *Lecture Notes in Computer Science*. Springer, 1997. 112–131
12. N.Gama, P.Q.Nguyen, Predicting lattice reduction. In: EUROCRYPT 2008. LNCS, vol.4965, pp.31-51. Istanbul, Turkey, 2008.
13. N.Gama, P.Q.Nguyen, Finding short lattice vectors within Mordell's inequality. In: STOC 2008. pp.207-216. Victoria, British Columbia, Canada, 2008.
14. N.Gama, P.Q.Nguyen, O.Regev, Lattice enumeration using extreme pruning. In EUROCRYPT 2010. LNCS, vol.6110, pp.257-278, French Riviera, 2010.
15. C. Gentry, C. Peikert and V. Vaikuntanathan, Trapdoors for hard lattices and new cryptographic constructions. In STOC 2008. pp.197-206, Victoria, British Columbia, Canada, 2008.
16. D.Han, M-H.Kim and Y.Yeom, Cryptanalysis of the Paeng-Jung-Ha cryptosystem from PKC 2003. In PKC 2007. LNCS,vol. 4450, pp.107-117, Beijing, China, 2007.
17. J.Hoffstein, J. Pipher, J.H.Silverman, NTRU: A ring-based public key cryptosystem. In: ANTS 1998. LNCS, vol. 1423, pp.267-288. Portland, Oregon, USA, 1998
18. G.Hanrot,X.Pujol,D.Stehlé, Algorithms for the shortest and closest lattice vector problems. In IWCC 2011. pp.159-190. Qingdao China 2011.
19. G.Hanrot,X.Pujol,D.Stehlé, Analyzing Blockwise Lattice Algorithms Using Dynamical Systems. In CRYPTO 2011. LNCS, vol.6841 ,pp.447-464, Santa Barbara, USA, 2011.

20. R.Kannan, Improved algorithms for integer programming and related lattice problems. In STOC 1983. pp.193-206. Boston, Massachusetts, USA, 1983.
21. R.Kannan, Minkowski's convex body theorem and integer programming. Mathematics of Operations Research, 12(3)(1987), pp.415-440.
22. G. Kabatiansky and V. Levenshtein. Bounds for packings on a sphere and in space. Problemy Peredachi Informatsii, 14(1)(1978), pp.3-25.
23. A. K.Lenstra, H. W.Jr.Lenstra, L.Lovász, Factoring polynomials with rational coefficients. Mathematische Analen 261(1982), pp.513-534.
24. C.Ling, S.Liu, L.Luzzi, D Stehlé, Decoding by embedding: correct decoding radius and DMT optimality. In ISIT 2011. pp.1106-1110. Saint-Petersburg Russia. 2011.
25. V.Lyubashevsky and D.Micciancio, On bounded distance decoding, unique shortest vectors, and the minnimum distance problem. In CRYPTO 2009. LNCS, vol.5677, pp.577-594. Santa Barbara, California, USA, 2009.
26. Liu M, Nguyen P Q. Solving BDD by enumeration: An update. In: Dawson E, eds. CT-RSA, Lecture Notes in Computer Science, vol. 7779, Springer, 2013. 293–309
27. J.C.Lagarias and A.M.Odlyzko, Solving low-density subset sum problems. Jounal of the Association for Computing Machinery, 32(1)(1985), pp229-246.
28. R.Lindner and C.Peikert. Better key sizes (and attacks) for LWE-based encryption. In: CT-RSA 2011, pp319-339, San Francisco, CA, USA, 2011.
29. D.Micciancio, Efficient reductions among lattice problems. In SODA 2008. pp.84-93. San Francisco, California, USA, 2008.
30. J.Milnorv and D.Husemoller, Symmetric Bilinear Forms. Math.Z(1973).
31. J.E.Mazo and A.M.Odlyzko, Lattice points in high-dimensional sheres. Monatsh. Math.,110(1990), pp.47-61.
32. D.Micciancio and O.Regev, Lattice-based Cryptography,in Post-Quantum Cryptography, Daniel J. Bernstein, Johannes Buchmann and Erik Dahmen, eds., Springer-Verlag,Berlin, 2009, pp.147-187.
33. D.Micciancio and P.Voulgaris, Faster exponential time algorithms for the shortest vector problem. In SODA 2010. pp. 1468-1480, Austin, Texas, USA, 2010.
34. D.Micciancio and P.Voulgaris, A Deterministic Single Exponential Time Algorithm for Most Lattice Problems based on Voronoi Cell Computations. In STOC 2010. pp.351-358. Cambridge, Massachusetts, USA, 2010.
35. P.Q.Nguyen, Cryptanalysis of the Goldreich-Goldwasser-Halevi cryptosystem from Crypto'97. In CRYPTO 1999. LNCS, vol.1666, pp.288-304. Santa Barbara, California, USA, 1999.
36. P.Q.Nguyen and T. Vidick, Sieve algorithms for the shortest vector problem are practical. J. of Mathematical Cryptology, 2(2)(2008), pp181-207.
37. C.Peikert, Public-key cryptosystems from the worst-case shortest vector problem. In STOC 2009. pp.333-342. Bethesda, MD, USA, 2009.
38. M.Pohst, On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications. ACM SIGSAM Bulletin 15(1)(1981), pp.37-44.
39. X.Pujol, and D.Stehlé, Solving the shortest lattice vector problem in time $2^{2.465n}$. Cryptology ePrint Archive, Report 2009/605(2009) Available at http://eprint.iacr.org/2009/605 .
40. T.Plantard and W.Susilo, Broadcast attacks against lattice-based cryptosystems. In ACNS 2009. LNCS, vol.5536, pp.456-472. Paris-Rocquencourt, France, 2009.
41. O.Regev, Lecture notes on lattices in computer science, 2004. Available at http://www.cs.tau.ac.il/ odedr/teaching/lattices fall 2004/index. html.
42. M.Ruckert1, M.Schneider. Estimating the Security of Lattice-based Cryptosystems. Cryptology ePrint Archive, Report 2010/137(2010) Available at http://eprint.iacr.org/2010/137 .
43. O.Regev, On lattices, learing with errors, random linear codes, and cryptography. J.ACM 56(6)(2009), pp1-40.
44. C.P.Schnorr, A hierarchy of Polynomial Lattice Basisi Reduction Algorithms. Theoretical Computer Science, 53(1987), pp201-224(1987).
45. C.P.Schnorr and M.Euchner, Lattice basis reduction: improved practical algorithms and solving subset sum problems. Mathematics of Programming 66(1994), pp.181-199.
46. D.Stehlé, R.Steinfeld, K.Tanaka, and K.Xagawa, Efficient public key encryption based on ideal lattices. In: ASIACRYPT 2009. LNCS, vol.5912, pp.617-635.Tokyo, Japan, 2009.
47. X.Y.Wang, M.J.Liu, C.L.Tian and J.G.Bi, Improved Nguyen-Vidick heuristic sieve algorithm for shortest vector problem. In : ASIACCS 2011, pp.1-9, Hongkong, China, 2011

7 Appendix

Proof of Theorem 2

Proof. By lemma 5, we only need to prove that there is a reduction from $\mathrm{BDD}_{\gamma_1(1-\frac{1}{n})}$ to $uSVP_\gamma$.

Suppose $u = dist(\mathbf{t}, \mathcal{L}) < \gamma_1(1-\frac{1}{n})\lambda_1(\mathcal{L})$, and $\mathbf{v} \in \mathcal{L}$ is a vector such that $dist(\mathbf{t}, \mathcal{L}) = \|\mathbf{v}-\mathbf{t}\|$. We shall show that the vector $\mathbf{v}' = (\mathbf{v}-\mathbf{t}, -ku_0)$ is the γ-unique shortest vector in $\mathcal{L}'$ and hence can be found by solving $uSVP_\gamma$. Then, we get $\mathbf{v}$ from $\mathbf{v}'$ immediately.

Note that $u \le u_0 < \frac{u}{1-\frac{1}{n}}$, so we have $u_0 < \gamma_1\lambda_1$.

To show that $\mathbf{v}'$ is the $\gamma-$unique shortest vector in $\mathcal{L}'$, it suffices to show that any vector $\mathbf{y} \in \mathcal{L}' \setminus \mathbb{Z}\mathbf{v}'$ has length bigger than $\gamma\lambda_1 = \gamma\|\mathbf{v}'\| = \gamma\sqrt{u^2+(ku_0)^2}$.

Now let us set $k = \frac{\gamma}{\sqrt{4-\gamma^2}}$. Notice that $\gamma_1 = \frac{\sqrt{4-\gamma^2}}{\sqrt{3}\gamma+\sqrt{4-\gamma^2}}$, we see that

$$\gamma\gamma_1\sqrt{1+k^2} = \frac{2\gamma}{\sqrt{3}\gamma+\sqrt{4-\gamma^2}}. \tag{2}$$

Write $\mathbf{y} = \Sigma_{i=1}^n x_i\mathbf{b}'_i + m\mathbf{b}'_{n+1}$. Then $\mathbf{y} = (\mathbf{a}+m\mathbf{t}, mku_0)$ for some $\mathbf{a} \in \mathcal{L}$. We divide the rest of our proof into three cases.

Case 1: $m = 0$. In this case we must have $\mathbf{a} \neq \mathbf{0}$ and $\|\mathbf{y}\| = \|\mathbf{a}\|$. From equation (2), $\gamma\gamma_1\sqrt{1+k^2} < 1$, we have $\|\mathbf{y}\| = \|\mathbf{a}\| \ge \lambda_1 > \gamma\sqrt{1+k^2}\gamma_1\lambda_1 > \gamma\sqrt{1+k^2}u_0 \ge \gamma\sqrt{u^2+k^2u_0^2}$.

Case 2: $|m| = 1$. We only treat the case of $m = 1$, as the case of $m = -1$ is similar. Since $y \notin \mathbb{Z}\mathbf{v}'$, we know that $\mathbf{a} \neq -\mathbf{v}$. We also have $\|\mathbf{a}+\mathbf{v}\| \ge \lambda_1 > u_0$, because $\gamma_1 < 1$. Since $\frac{1}{\gamma_1} - 1 = \frac{\sqrt{3}\gamma}{\sqrt{4-\gamma^2}} = \sqrt{\gamma^2(1+k^2)-k^2}$, we have

$$\begin{aligned}\|\mathbf{y}\| &= \sqrt{\|\mathbf{a}+\mathbf{t}\|^2+k^2u_0^2} \ge \sqrt{(\|\mathbf{a}+\mathbf{v}\|-\|\mathbf{v}-\mathbf{t}\|)^2+k^2u_0^2} = \sqrt{(\|\mathbf{a}+\mathbf{v}\|-u)^2+k^2u_0^2}\\ &\ge \sqrt{(\lambda_1-u_0)^2+k^2u_0^2} > \sqrt{(\frac{u_0}{\gamma_1}-u_0)^2+k^2u_0^2} = \sqrt{(\frac{1}{\gamma_1}-1)^2u_0^2+k^2u_0^2}\\ &= \sqrt{(\gamma^2(1+k^2)-k^2)u_0^2+k^2u_0^2} = \gamma\sqrt{(1+k^2)u_0^2} \ge \gamma\sqrt{u^2+k^2u_0^2}\end{aligned}$$

Case 3 $|m| \ge 2$. Without loss of generality, we assume $2\mathbf{t} \notin \mathcal{L}$. Since $\gamma\sqrt{1+k^2} = 2k$, we have $\|\mathbf{y}\| = \sqrt{\|a+mt\|^2+m^2k^2u_0^2} > 2ku_0 = \gamma\sqrt{(1+k^2)u_0^2} \ge \gamma\sqrt{u^2+k^2u_0^2}$. The proof is complete.

Proof of Lemma 2

Proof. In this case, we have an exact SVP oracle, i.e.$\gamma = 1$. As in the proof of Theorem 2, let $k = \frac{\gamma}{\sqrt{4-\gamma^2}} = \frac{\sqrt{3}}{3}$ and discuss three cases. But, here we take $\gamma_1 = \frac{\sqrt{3}}{2}$. For case 1 of Theorem 2, it is obvious that

$$\|\mathbf{y}\| = \|\mathbf{a}\| \ge \lambda_1 > \frac{2}{\sqrt{3}}u_0 = \sqrt{1+k^2}u_0 \ge \sqrt{u^2+k^2u_0^2}.$$

The case 2 of Theorem 2 can be adapted to $\|(\mathbf{a}-\mathbf{t}, -ku_0)\| > \sqrt{u^2+k^2u_0^2}$. If $\mathbf{a}$ is not a closest lattice vector to $\mathbf{t}$, this inequality is trivial. The analysis of case 3 is the same as in Theorem 2.

This proves that $BDD_{\frac{\sqrt{3}}{2}}$ reduces to SVP.

Next we move to the proof of Lemma 10. We need the following lemma in our proof.

Lemma 12. *[22] Let $E \subseteq \mathbb{R}^n \backslash \{\mathbf{0}\}$. 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 10

Proof. 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 $\|\mathbf{t}'\| \leq 2^{O(n)}\mu$. After the first loop of the algorithm, we have $c\mu \leq \|\mathbf{t}_i\| \leq (2^{O(n)} + \xi)\mu$. We 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

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

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 Lemma 12 we get the conclusion.