Hardness Estimation of LWE via Band Pruning [*]

Yoshinori Aono Le Trieu Phong Lihua Wang [†]

October 23, 2015

Abstract

This paper, examining the hardness of the search LWE problem, is a refined continuation of previous works including (Lindner-Peikert 2011, Liu-Nguyen 2013, Aono et al. 2013) using lattice reduction and lattice vector enumeration. We adopt the attack to the LWE using discrete Gaussian distribution, and propose a new bounding method named band pruning in lattice enumeration. We update the security estimations for several parameter sets proposed in the literature. Finally, using the data gained in our experiments, we derive an explicit formula linking the LWE's parameters with the bit security.

Keywords: Search LWE, hardness, lattice reduction, lattice enumeration

1 Introduction

1.1 Background

Fix numbers n and q. The learning with errors (LWE) problem [23] is roughly the problem to find a secret vector $x \in \mathbb{Z}^n$ from a set of samples $(a_i \in \mathbb{Z}^n, b_i \in \mathbb{Z})$ where

$$\langle a_i, x\rangle + e_i = b_i \text{ (mod } q) \text{ for } i = 1, 2, \ldots$$

where a_i is randomly sampled from $\{0, \ldots, q-1\}^n$ and e_i is sampled from a distribution over a set of small integers of deviation s. (The detailed definition is in Section 2.3)

The hardness of the learning with errors problem is a gold mine for cryptographers. It has been widely used to ensure the security of numerous cryptographic schemes. Therefore, giving concrete parameters for the hardness is a must in practice.

1.2 Our Contributions

This paper is an update of lattice vector enumeration based analyses [6, 18] for the LWE problem, and gives security analyses for several proposed parameter sets [6, 17, 18]. In details, our technical contributions are as follows:

(1) we update (refine) the cost estimation of lattice reduction part using recent records in SVP challenge [2]. Besides the time for lattice reduction, we need to predict the lengths of Gram-Schmidt basis vectors as sharp

[*] An abridged part of this paper was presented in [6] as the analytic contribution. This manuscript extends that part significantly.

[†] The authors are with Network Security Research Institute, National Institute of Information and Communications Technology, Japan.

as possible to predict the cost of lattice point search and consider the trade-off between two timings. We give our new estimation under both Lindner-Peikert model [17] and Chen-Nguyen [12] model with modified coefficients.

(2) we develop a new pruning method, which we will refer as *the band pruning*, to speed up the lattice vector enumeration. Our theoretical analysis considers *discrete* Gaussian error vectors while the previous attacks [17, 18] only consider continuous Gaussian ones[1], allowing more rigid security analysis using small (e.g., deviation less than 8.0) Gaussian parameters.

(3) we give a new method to estimate the lattice vector enumeration cost, which is derived from the volume of a bit complicated n-dimensional object (see Seciton 4.) To approximate the volume, we use a method inspired from Gama-Nguyen-Regev [13] in the conference version [6] though it was omitted due to the space limit. After the conference version, we find a new method to approximate the volume without using a random source, which was a drawback of the original method. For the completeness of the information, we give both methods in Section 4.1 and A.

(4) combining these new techniques together, we give security estimations for several parameters and success probabilities. By curve fitting on the data gained in our experiments, we derive an explicit formula to link between the bit security and LWE's parameters (n, q, s) as follows:

$$\textbf{bit}-\textbf{security} = \frac{7.18n - 219}{\ln(q) - 1.66\ln(s)} - 87 \tag{1}$$

The left hand side **bit-security** is defined as follows:

$$\textbf{bit}-\textbf{security} \text{ of LWE}(n, q, s) = \log_2\left(\frac{\text{attacking time in seconds}}{\text{attack success probability}}\right) + \log_2(9 \cdot 10^6) \tag{2}$$

Here, the constant $\log_2(9 \cdot 10^6)$ to convert time in seconds to bit-security is from the result of `RC5-72` benchmark published in `distributed.net`. It makes a record that a standard Intel CPU can check about $9 \cdot 10^6$ keys in second in one thread.

1.3 Discussion on the Possibility of Optimizing Bounding Function

As the cases to estimate SVP and CVP, we simulate the cost and success probability p_{succ} when we search an area defined by the bounding function. It is clear that the optimized bounding function that achieves minimizing cost with keeping some probability, gives the hardness of cryptographies. Several fast computing methods to approximate the cost and p_{succ} have been developed when we assume the target point distributes uniformly in the searching area, and it allows us to optimize bounding function.

On the other hand, we assume the error vector distributes as a discrete Gaussian in this paper. This discreteness makes the estimation of probability very complicated and the problem of optimizing bounding function becomes a practically hard problem. Efficient generation of optimal function is an interesting future work.

1.4 Related Works

The LWE problem in dimension n can be theoretically reduced to lattice problems in dimension $\sqrt{n}$ as in [10]. On the practical side, to give the concrete security parameters, several attacks are proposed which

[1] Quoting from [17, Section 6]: "... to allow for a Gaussian parameter $s \geq 8$, so that the discrete Gaussian $D_{\mathbb{Z}_m,s}$ approximates the continuous Gaussian D_s extremely well."

are mainly classified to three types. Since the polynomial-time equivalence between the decision and search versions [23], many known attacks consider the search version while the securities of schemes are from the decision version. A survey on recent algorithms for solving LWE is in Albrecht et al. [5].

Lattice-Based attacks: Micciancio and Regev [20] gave a distinguishing attack using lattice reduction to a gadget lattice basis of which the first vector of a reduced basis corresponds to the error vector. In this line, Bai-Galbraith [8] and Lauter et al. [16] investigated the analysis.

Subsequently, Lindner-Peikert [17] regarded the problem to find the error vector as the bounded distance decoding (BDD) and analyzed a randomized version of the Babai's nearest plane algorithm. Liu-Nguyen [18] also considered the problem as BDD and estimated the computational cost by lattice enumeration with linear pruning. These works assume the noises are from a continuous Gaussian distribution. The attack in Aono et al. [6] improved all these attacks by viewing the LWE problem as the closest vector problem (CVP) in which the difference between the target vector and lattice vectors has a discrete Gaussian distribution.

When the Gaussian error is continuous, it is the problem to recover the received signals in MIMO wireless connection. Several lattice based algorithms have been proposed [19, 26].

BKW attacks: Because the LWE problem is a natural extension of the learning with parity noises (LPN) problem, namely, LPN is LWE with $q = 2$, algorithms for LPN problem can be adopted for solving LWE problems. While Blum-Kalai-Wasserman [9] was originally proposed to solve the LPN problem, it was imported to the cryptographic area [3] and has been deeply investigated. The early attacks have a drawback that requires exponential number of samples, that does not match the real scheme, whereas the problem is avoided by considering trade-off between the complexity and number of samples.

Algebraic attacks: This type of LWE attacking algorithm is converting the original problem to an algebraic equation over integers. Arora-Ge [7] proposed a method to convert the problem to a large dimensional linear equation. The algorithm was further improved by the Gröebner basis technique [4].

1.5 Roadmap

We give necessary lemmas and theorems, and introduce several previous works in Section 2. We fix the models and give our cost and probability estimation for the lattice reduction part in Section 3 and for the lattice enumeration part in Section 4. In Section 5, we propose a method to set our bounding function used in lattice enumeration. Finally, we give our new estimation for LWE problem and concrete formula in Section 6.

2 Preliminaries

Throughout this paper, we use $\log_2$ and ln to denote the logarithms of base 2 and of natural base.

2.1 Lattices

For a set of linearly independent vectors $(v_1, \dots, v_m)$, which is called *a lattice basis*, *the lattice* is set $L := \left\{\sum_{i=1}^m a_i v_i : a_i \in \mathbb{Z}\right\}$. We denote its Gram-Schmidt basis by tildes: $\tilde{v_1}, \dots, \tilde{v_m}$. *The lattice volume or determinant* is $\det(L) := \prod_{i=1}^m \|\tilde{v_i}\|$. For a fixed lattice basis $(v_1, \dots, v_m)$ and a vector $v = \sum_{i=1}^m x_i \tilde{v_i}$, its k-th projection is $\pi_k(v) = \sum_{i=k}^m x_i \tilde{v_i}$. For lattice theories used in cryptographic area, see [22].

2.2 Discrete Gaussian

For the derivation parameter $s > 0$, the discrete Gaussian distribution ψ_s over $\mathbb{Z}$ is the random variable whose density function at y is

$$\Pr[\psi_s = y] = \frac{\exp(-\pi y^2/s^2)}{1 + 2\sum_{j=1}^{\infty} \exp(-\pi j^2/s^2)}. \tag{3}$$

Denote the denominator in (3) as $W(s)$. The m-dimensional discrete Gaussian $e = (e_1, \dots, e_m) \in \psi_s^m$ is defined by taking each e_i from ψ_s independently. Thus, we have

$$\Pr[\psi_s^m = (e_1, \dots, e_m)] = \frac{\exp(-\pi \|e\|^2/s^2)}{W(s)^m}. \tag{4}$$

To treat the discrete Gaussian, the following lemma is necessary.

Lemma 1 *For given $m, B \in \mathbb{Z}$ and small $s \in \mathbb{R}$, we can efficiently compute the probability*

$$p(m, B) := \Pr_{e \leftarrow \psi_s^{m\times 1}} \left[\|e\|^2 = B \right] = \frac{\displaystyle\sum_{\substack{y \in \mathbb{Z}^m \\ \|y\|^2 = B}} \exp(-\pi \|y\|^2/s^2)}{W(s)^m}$$

with high accuracy.

Proof. Consider the sequence $\{p(1, i)\}_{i=0,1,\dots}$ that is easily computed with high accuracy. By the relation $p(m+1, B) = \sum_{j=0}^{B} p(m, j) \cdot p(1, B - j)$, $\{p(m+1, i)\}_{i=0,1,\dots}$ is the convolution of $\{p(m, i)\}_{i=0,1,\dots}$ and $\{p(1, i)\}_{i=0,1,\dots}$. Considering an aborted sequence $\{p(1, i)\}_{i=0,1,\dots,N-1}$ of length N of some power of two, the convolution can be efficiently computed by the FFT. □

In this paper, it is enough by computing these values in 512-bit precision because we will argue the parameters that achieve at most about 256-bit security. If one wants to obtain these values in very high accuracy, compute its numerator $\eta(B, m) \exp(-\pi B/s^2)$ where $\eta(B, m)$ is the number of integer solutions of the Diophantine equation $x_1^2 + \cdots + x_m^2 = B$. This is just the coefficient of x^B of a special case of the Jacobi theta function $(\sum_{j=-\infty}^{\infty} x^{j^2})^m = (1 + 2\sum_{j=1}^{\infty} x^{j^2})^m$. The coefficients are easily derived by computing this function modulo x^ℓ for some ℓ.

With the same method, the probability that $\|e\|$ is within some range is also computable:

$$\Pr_{e \leftarrow \psi_s^{m\times 1}} \left[B_1 \le \|e\|^2 \le B_2 \right] = \sum_{B \in [B_1, B_2] \cap \mathbb{Z}} \Pr_{e \leftarrow \psi_s^{m\times 1}} \left[\|e\|^2 = B \right].$$

2.3 The (Search) Learning With Errors Problem

The search LWE is defined as follows. For given ($A \in \mathbb{Z}^{m\times n}$, $b = Ax + e \in \mathbb{Z}^{m\times 1}$) where $x \in \psi_{\alpha q}^{n\times 1}$ and $e \in \psi_{\alpha q}^{m\times 1}$, the problem is to find the secret vector x or equivalently to find e. Let us denote as LWE(n, α, q).

For given instance (A, b), the standard lattice based attack considers the lattice $\Lambda_q(A) := \{z \in \mathbb{Z}^m : \exists x \text{ such that } z = Ax \pmod{q}\}$ whose vector z closest to b derives the error vector $e = b - z$. Assume the lattice $\Lambda_q(A)$ is given by rows of a q-ary matrix:

$$\begin{bmatrix} qI_{m-n} & 0 \\ A' & I_n \end{bmatrix}$$

Reference	Predicting $\|\mathbf{b}_i^*\|$	Computing Time
[17]	GSA	$\log_2(Time_{BKZ}(\delta)) = 1.8/\log_2(\delta) - 110$
[12]	GSA	Upper bound of ENUM cost [12, Table 4]
Our model 1 (Section 3.1)	GSA	$\log_2(Time_{BKZ}(\delta)) = 1.8/\log_2(\delta) - 130$
Our model 2 (Section 3.2)	Chen-Nguyen's simulator	$\log_2(\mathrm{Cost}_\beta) = \alpha\log_2(L_\beta) + (1-\alpha)\log_2(U_\beta)$ ($\alpha = 0.2$)

Table 1: Summery of attacking models among previous works and this paper

where A' is uniquely determined under modulo q from the instance. We use $(b_1, \ldots, b_m)$ and $(\tilde{b}_1, \ldots, \tilde{b}_m)$ to denote a reduced basis and its Gram-Schmidt basis. As we will introduced later, the cost of lattice enumeration part can be approximated using only the Gram-Schmidt lengths $(\|\mathbf{b}_1^*\|, \ldots, \|\mathbf{b}_m^*\|)$ and bounding coefficients.

2.4 Experimenting Environment

We used the boost library [1] to compute the bounding functions in lattice vector enumeration, and to compute the attacking cost and success probability. The preliminary experiments in Section 3 was performed with using ntl library [27].

3 Models for Lattice Reduction

To analyse the lattice based attack for LWE, it needs to fix the model for lattice reduction part. Concretely, the lengths $\|\widetilde{b_i}\|$ of Gram-Schmidt basis which is used to estimate the lattice vector enumeration part, and time for lattice reduction. Following the previous works [17, 18], we consider two models, the geometric series assumption (GSA) model [24] and BKZ 2.0 model [12] with modifying constant factor using Lattice Challenge records [2]. Table 1 shows the summery, and below we give the details of them.

3.1 Geometric Series Assumption Model

Since the target lattice is q-ary, following Schnorr [24] and experiments in [17], we set the following assumption here.

Assumption 1 *The graph of* $\ln\|\widetilde{b_i}\|$ *consists of horizontal line* $\|\widetilde{b_i}\| = q$ *(if they exist), slope of* $0.5\ln r$*, and line* $\|\widetilde{b_i}\| = 1$.

Here, r is a constant in GSA that assumes $\|\widetilde{b_i}\|^2/\|b_1\|^2 = r^{i-1}$ for a reduced basis. It connects to the root Hermite factor δ by the relation $r = \delta^{-4m/(m-1)}$ if the Gram-Schmidt basis lengths consist only the slope part. The condition is explicitly given as follows.

$$1 < \|\widetilde{b_i}\| < q \text{ for all } i \in [m] \;\Leftrightarrow\; \left(\ln q - \sqrt{\ln^2 q - 4n\ln\delta\ln q}\right)/2\ln\delta < m < \sqrt{(n\ln q)/\ln\delta} \tag{5}$$

In this situation, δ is an algorithm-depended factor so that $\|b_1\| = \delta^n\det(L)^{1/n}$ holds.

Computing time: In [17], they estimated the cost of BKZ algorithm to achieve the root Hermite factor δ as

$$Time_{LP}(\delta) = 2^{1.8/\lg(\delta)-c} \text{ [single-core seconds]},$$

where they used $c = 110$. Figure 1 plots recent records in SVP Challenge. From the figure, we decided to use $c = 130$ as the average performance of recent algorithms.

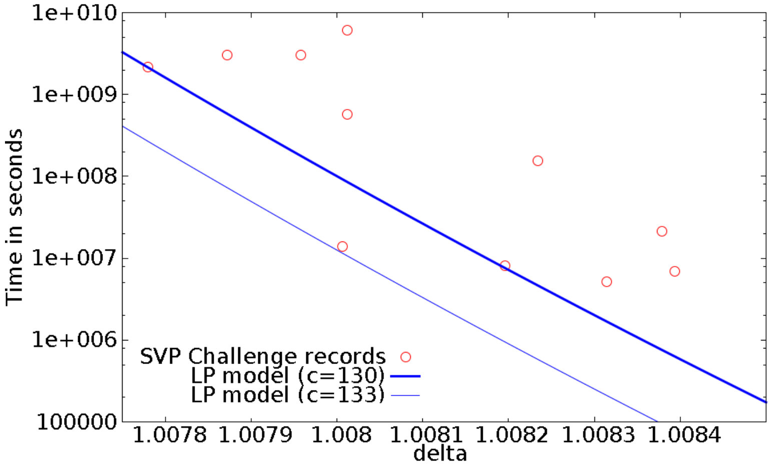

Figure 1: Relation between root Hermite factor and computing time published in SVP Challenge, and modified Lindner-Peikert estimation..

Drawback of this model: Clearly, there exists a lower bound of δ from the lengths of the shortest vector. We can see that substituting its value, $Time_{LP} = 2^{\Theta(n/\ln(n))}$. Thus, it derives a subexponential algorithm for SVP, which is probably hard to realize. For detailed argument, see Albrecht et al. [5].

Moreover, it is known that Schnorr's GSA does not hold in general when the basis is very reduced [11]. Remark that Liu-Nguyen [18] also employed this model with modified computing time from Chen-Nguyen's BKZ 2.0 simulator.

To avoid these drawbacks, we use the BKZ 2.0 Model.

3.2 BKZ 2.0 Model

To modify the drawback of GSA model., we use Chen-Nguyen's estimation and simulator.

From [12, Table 4,5], we extrapolate the lower and upper bound of the number of processed nodes in one enumeration of BKZ-β as

$$\begin{aligned}\log_2(U_\beta) &= 0.000784314\beta^2 + 0.366078\beta - 6.125, \text{ and}\\ \log_2(L_\beta) &= 0.000409753\beta^2 + 0.237652\beta - 19.3668.\end{aligned}$$

Since they have a significant gap, we need to select a good medium estimation to meet the experimental data. For this purpose, we assume the cost of enumeration satisfies

$$\log_2(\text{Cost}_\beta) = \alpha \log_2(L_\beta) + (1-\alpha)\log_2(U_\beta), \tag{6}$$

and for several fixed $\alpha \in [0, 1]$ and lattice dimension, we execute their BKZ simulator to solve the SVP challenge problems and search the optimal β minimizing total enumeration cost. The simulation starts at the simulated LLL-reduced basis that satisfies GSA and $\delta = 1.022$ whose constant is from [21]. Total enumeration cost in seconds is

$$\mathrm{Time}_{\mathrm{BKZ2.0}}(n, \beta, \sharp\mathrm{Tours}) = \sharp\mathrm{Tours} \cdot \sum_{i=1}^{n-1} \mathrm{Cost}_{\min(\beta, n-i+1)}/(5.0 \cdot 10^7)[\mathrm{sec}] \tag{7}$$

Comparing with the recent records (see Figure 2), we decide that $\alpha = 0.2$ gives the practical lower bound to the lattice reduction cost at state of the art, and we will use in this paper. Here, the constant $5.0 \cdot 10^7$ is decided from our benchmark on lattice enumeration.

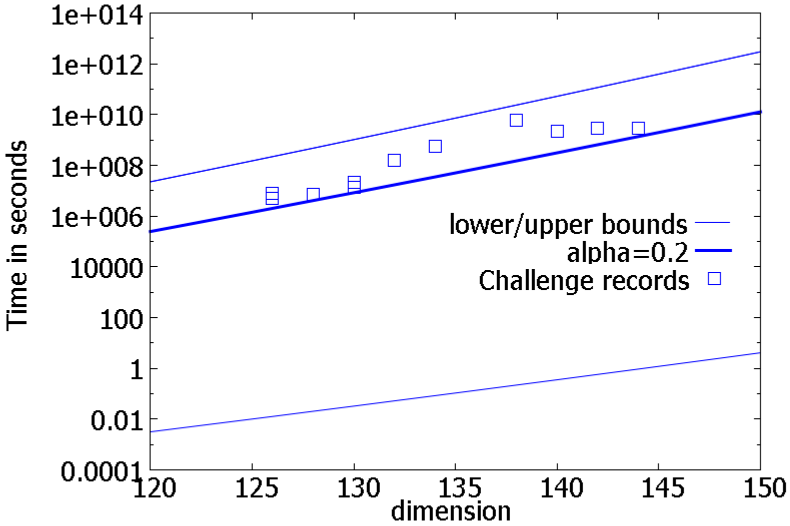

Figure 2: Costs for solving SVP Challenge problems simulated by Chen-Nguyen's BKZ 2.0 simulator. Lower and upper bounds, costs when $\alpha = 0.2$ and points from [2].

4 Model for Lattice Vector Enumeration

We introduce our model for searching error vector, success probability and computing cost adopting to the discrete Gaussian model.

Lattice vector enumeration algorithm: To find the error vector, our attack employs an exhaustive search algorithm [15] (and its modifications [18, 25]) with pruning technique adopted for discrete Gaussian LWE. Since the coordinates of error vector are from a Gaussian, we can bound its projected lengths from lower and upper.

The outline of algorithm is as follows: Consider a search tree whose root is labeled by the vector b. For each node labeled by v at depth k, its children have labels of the form $v - a \cdot b_{m-k}$ with $a \in \mathbb{Z}$. Thus, nodes at depth k are labeled by vectors of the form $b - \sum_{i=m-k+1}^{m} a_i \cdot b_i \in b - \Lambda_q(A)$ and the desired vector e exists at depth m. The algorithm is the depth-first search for this tree and a node is pruned if the projected length $\|\pi_{m-k+1}(b - \sum_{i=m-k+1}^{m} a_i \cdot b_i)\|$ is outer of the range $[L_k, R_k]$ which we will give later.

Assumptions to set the bonding coefficients:

Assumption 2 *[13, Heuristic 3] the distribution of matrix* $(\tilde{b}_1/||\tilde{b}_1||, \dots, \tilde{b}_m/||\tilde{b}_m||)$ *of a random reduced basis looks like a uniform distribution over* $R_O^{m\times m}$ *in the meaning of Haar measure, the set of normalized orthogonal matrix of degree m. In particular, for a fixed point* $w \in S^{m-1}$*,* wV *distributes uniformly over* S^{m-1} *when* $V \leftarrow R_O^{m\times m}$*.*

Note that this assumption does not hold in general q-ary lattices, because for some reduced bases and parameters, $\widetilde{b_i}$'s for small and large indexes remain as $(0, \dots, 0, q, 0, \dots, 0)$ and $(0, \dots, 0, 1, 0, \dots, 0)$, respectively. To avoid this phenomenon, we restricted the range of m and δ by (5) in parameter setting.

From now on, denote $v_i := \widetilde{b_i}/||\widetilde{b_i}||$. For $b = Ax + e$, write the error vector $e := b - z = (e_1, \dots, e_m) = \sum_{i=1}^m \alpha_i \widetilde{b_i}$ with $z = Ax \in \Lambda_q(A)$. By $\langle e, \widetilde{b_i}\rangle = \alpha_i ||\widetilde{b_i}||^2$, the projective length of each node is

$$\left\|\pi_{m-k+1}\Big(b - \sum_{i=m-k+1}^{m} a_i \cdot b_i\Big)\right\|^2 = \sum_{i=m-k+1}^{m} \alpha_i^2 ||\widetilde{b_i}||^2 = \sum_{i=m-k+1}^{m} \langle e, \widetilde{b_i}/||\widetilde{b_i}||\rangle^2 = \sum_{i=m-k+1}^{m} \langle e, v_i\rangle^2.$$

We discuss the distribution of this length when $(v_1, \dots, v_m) \leftarrow R_O^{m\times m}$ and $e \leftarrow \psi_s^{m\times 1}$. Let $V = (v_1, \dots, v_m)$. $\langle v_i, v_j\rangle = \delta_{ij}$ and $V^{-1} = V^T$ hold. Since $\langle e, v_i\rangle = \langle V^{-1}e, V^{-1}v_i\rangle = (V^{-1}e)_i$ (the i-th element of vector) and $||V^{-1}e|| = ||e||$, the distribution of $\sum_{i=1}^k \langle e, v_i\rangle^2$ is the same as that of $||e||(g_1^2 + \cdots + g_k^2)$ where $(g_1, \dots, g_m) \overset{\$}{\leftarrow} S^{m-1}$. In other words, the norm distribution is unchanged, whereas its position distributes over the scaled $(n-1)$-sphere. We denote this distribution $C_{s,m}$. This is our model of the distribution of error vector.

Definitions of cost and probability:

For fixed parameters and bounding functions L_i and R_i, we define the success probability of the attack as follows.

Definition 1 *Success probability of the attack.*

$$\Pr\Big[L_k^2 < \sum_{i=1}^{k} \langle e, v_i\rangle^2 < R_k^2 \ \forall k \in [m]\Big]. \tag{8}$$

Here, the probability is over $e \leftarrow \psi_s^{m\times 1}$ *and* $(v_1, \dots, v_m) \leftarrow R_O^{m\times m}$*.*

The above can be decomposed as

$$\Pr_{f\leftarrow C_{s,m}}\Big[L_k^2 < \sum_{i=1}^{k} f_i^2 < R_k^2 \ \forall k \in [m]\Big] = \sum_{u=L_m^2}^{R_m^2} \Pr_{f\leftarrow C_{s,m}}\Big[\sum_{i=1}^{k} f_i^2 \in [L_k^2, R_k^2] \ \forall k \in [m] \Big| ||f||^2 = u\Big] \times \Pr_{f\leftarrow C_{s,m}}[||f||^2 = u].$$

Remark that the latter factor $\Pr_{f\leftarrow C_{s,m}}[||f||^2 = u]$ is computable by using Claim 1, and we compute the other part by the Monte-Carlo sampling over sphere.

Following [13], by the Gaussian heuristic assumption, the approximated number of processed nodes during lattice enumeration is computable. We will use it to estimate the cost of lattice vector enumeration part.

Definition 2 *Cost of lattice vector enumeration.*

$$\sharp ENUM = \sum_{k=1}^{m} \frac{\mathrm{Vol} C(L_1,\dots,L_k;R_1,\dots,R_k)}{\prod_{i=m-k+1}^{m} \|\widetilde{b_i}\|}.$$

Here, $C(L_1,\dots,L_k;R_1,\dots,R_k)$ is the object defined by

$$\left\{(x_1,\dots,x_k)\in\mathbb{R}^k : L_i^2 < \sum_{\ell=1}^{i} x_\ell^2 < R_i^2 \text{ for } \forall i\in[k]\right\}.$$

To find a good cost estimation, we need to approximate the volume of this object.

4.1 Approximating Volume Factors

Fix an integer $k \geq 1$. Let the sequences $(L_1,\dots,L_k)$ and $(R_1,\dots,R_k)$ are monotonic increasing and satisfy $0 \leq L_i < R_i \leq 1$ for all i. We can assume $R_i \leq 1$ without loss of generality; if not, we normalize the sequence by deviding R_k. For simplicity, we use l_k and r_k to denote the sequence respectively. Then denote the object $C(l_k, r_k) := C(L_1,\dots,L_k;R_1,\dots,R_k)$.

To find a good approximation of $\mathrm{Vol}C(l_k, r_k)$, in the conference version [6], we used the random sampling method inspired from Gama-Nguyen-Regev's analysis for lattice vector enumeration [13]. Because this method requires the random source, the estimated volume is perturbated in each execution for the same parameters. It is a barrier to search optimized parameters correctly. Moreover, it requires a heavy computing when the dimension is high. To avoid the drawbacks, we develop a new approximating method. Although the old algorithm is not used in this paper, we give it in Appendix A for completeness of information because it was omitted in the conference proceeding.

Our method in theory: Since $C(l_k, r_k) \subset [-1,1]^k$, the volume can be written by the probability $P_k = \Pr_{x\leftarrow[-1,1]^k}[x \in C(l_k, r_k)]$ times 2^k.

For a point x in an Euclidean space of dimension $\geq j$, we denote the event E_j be that x satisfies $L_j^2 < \sum_{\ell=1}^{j} x_\ell^2 < R_j^2$. The desired probability is $\Pr_{x\in[-1,1]^k}[E_1\cdots E_k]$. For $i = 1,\dots,k$, we let

$$P_i = \Pr_{x\leftarrow[-1,1]^i}[E_i | E_1 \cdots E_{i-1}]$$

and we have by the chain rule

$$\begin{aligned} P_k &= \Pr_{x\leftarrow[-1,1]^k}[E_k|E_1\cdots E_{k-1}] \cdot \Pr_{x\leftarrow[-1,1]^k}[E_1\cdots E_{k-1}] \\ &= \Pr_{x\leftarrow[-1,1]^k}[E_k|E_1\cdots E_{k-1}] \cdot \Pr_{x\leftarrow[-1,1]^{k-1}}[E_1\cdots E_{k-1}] \\ &= \Pr_{x\leftarrow[-1,1]^k}[E_k|E_1\cdots E_{k-1}] \cdot P_{k-1}. \end{aligned}$$

We compute the probability by induction starting with the base case $P_1 = R_1 - L_1$.

By definition, the conditional probability is

$$\alpha_k := \Pr_{x\leftarrow[-1,1]^k}[E_k|E_1\cdots E_{k-1}] = \Pr_{x\leftarrow C(l_{k-1},r_{k-1})\times[-1,1]}[E_k]. \tag{9}$$

Denote $F_{k-1}(z)$ and $G_k(z)$ be the probability density function (p.d.f.) of $|x|^2$ and $|y|^2$ when $x \overset{\$}{\leftarrow} C(l_{k-1}, r_{k-1}) \subset \mathbb{R}^{k-1}$ and $y \overset{\$}{\leftarrow} C(l_{k-1}, r_{k-1}) \times [-1, 1] \subset \mathbb{R}^k$, respectively. It is clear that the distribution of $|y|^2$ is that of $|x|^2 + (x')^2$ where x is the same as above and $x' \leftarrow [-1, 1]$ independently. Thus, we have the relation

$$G_k(z) = F_{k-1}(z) * H(z) := \int_0^1 F_{k-1}(y)H(z-y)dy. \tag{10}$$

where

$$H(z) = \begin{cases} 1/2\sqrt{z} & (0 < z < 1) \\ 0 & (\text{otherwise}) \end{cases}$$

is the p.d.f. of x^2 when $x \overset{\$}{\leftarrow} [-1, 1]$.

By the relation $C(l_k, r_k) = C(l_{k-1}, r_{k-1}) \times [-1, 1] \cap \{x \in \mathbb{R}^k | L_k^2 \le |x|^2 \le R_k^2\}$, the probability (9) can be computed by

$$\Pr_{x \leftarrow C(l_{k-1}, r_{k-1}) \times [-1,1]}[L_k^2 \le |x|^2 \le R_k^2] = \int_{L_k^2}^{R_k^2} G_k(z)dz. \tag{11}$$

Also, we have the relation between the p.d.f.

$$F_k(z) = \begin{cases} (1/\alpha_k)G_k(z) & (L_k^2 \le z \le R_k^2) \\ 0 & (\text{otherwise}) \end{cases} \tag{12}$$

Therefore, the probability is

$$P_k = \prod_{i=1}^{k} \alpha_i.$$

Our method in practice: Our algorithm to compute the volume approximates the p.d.f. $F_k(z)$ and $G_k(z)$ within the range $[0, 1]$ by a real number sequence $f_j = (f_{j,0}, \ldots, f_{j,N-1})$ of length N where $f_{j,\ell}$ is an approximation for

$$\int_{(\ell-0.5)/N}^{(\ell+0.5)/N} F_j(x)dx. \tag{13}$$

The sequences $g_j = (g_{j,0}, \ldots, g_{j,N-1})$ and $h = (h_0, \ldots, h_{N-1})$ are also used for $G_j(x)$ and $H(x)$ in (13).

We simulate the theoretical argument as follows. For the base case, simulate

$$F_1(x) = \begin{cases} 2\sqrt{(R_1 - L_1)z} & (L_1^2 \le z \le R_1^2) \\ 0 & (otherwise) \end{cases}$$

and $H(z)$.

To simulate the convolution (10) of functions, we use the convolution of sequences $\{f_j * g_j\}_\ell = \{\sum_{i=0}^{\ell} f_{j,i} g_{j,\ell-i}\}_\ell$ which can be efficiently computable by using FFT.

The integral at the right-hand side of (11) is simulated by a simple addition:

$$\widetilde{\alpha_k} = \sum_{i=\ell_1}^{\ell_2} g_i$$

with $\ell_1 = [L_k^2 \cdot N]$ and $\ell_2 = [R_k^2 \cdot N]$. The cut-off (12) is multiply-then-zeroing:

$$f_{k,\ell} = \begin{cases} g_{k,\ell}/\widetilde{\alpha_k} & (\ell = \ell_1, \ldots, \ell_2) \\ 0 & (\text{otherwise}) \end{cases}$$

To simulate the values with sufficient accuracy, we selected $N = 2^{16}$ and used 150-bit precision floating point variables using the `boost` library.

5 LWE Hardness Estimation

In the rest of this section, the notation Pr without indicating distribution means that the probability over $f \leftarrow C_{s,m}$.

5.1 Our Bounding Function Setting

The goal of this section is to give a constructive proof of the following theorm.

Theorem 1 (Band pruning) *For any probability parameter $p \in [1/m, 1)$, under our model, we can efficiently compute numbers L_k and R_k so that the success probability is larger than $1 - p$.*

Proof. We again denote the event E_k be $L_k^2 < \sum_{i=1}^k f_i^2 < R_k^2$ holds and $\bar{E}_k$ for its inverse. Then the probability that the error vector is found is $p_{succ} := \Pr[E_1 \cdots E_m]$.

From Lemma 1 it is possible to compute the lower and upper bounds of error vector lengths L_m and R_m so that

$$\Pr_{e \overset{g}{\leftarrow} \mathbb{Z}_s^m}\left[\|e\|^2 > R_m^2\right] \le \frac{1}{2m} \text{ and } \Pr_{e \overset{g}{\leftarrow} \mathbb{Z}_s^m}\left[\|e\|^2 < L_m^2\right] \le \frac{1}{2m}.$$

Using these values, we have

$$p_{succ} = \Pr[E_m] \cdot \Pr[E_1 \cdots E_{m-1}|E_m] \ge \left(1 - \frac{1}{m}\right) \cdot \Pr[E_1 \cdots E_{m-1}|E_m].$$

To bound the probability factor, we consider the individual probability $\Pr[E_k|E_m]$. For any $L_k < R_k$, we have

$$\Pr[\bar{E}_k|E_m] = \sum_{u \in [L_m^2, R_m^2] \cap \mathbb{Z}} \Pr\Big[\sum_{i=1}^k f_i^2 \le L_k^2 \Big| \|f\|^2 = u\Big] \Pr\Big[\|f\|^2 = u\Big] + \sum_{u \in [L_m^2, R_m^2] \cap \mathbb{Z}} \Pr\Big[\sum_{i=1}^k f_i^2 \ge R_k^2 \Big| \|f\|^2 = u\Big] \Pr\Big[\|f\|^2 = u\Big]. \tag{14}$$

Each conditional probability can be represented by the incomplete beta function. For instance, the case of lower bound is

$$\Pr\Big[\sum_{i=1}^k f_i^2 \le L_k^2 \Big| \|f\|^2 = u\Big] = \Pr_{(h_1,\ldots,h_m) \leftarrow S^m}\Big[\sum_{i=1}^k h_i^2 \le \frac{L_k^2}{u}\Big] = I_{L_k^2/u}\left(\frac{k}{2}, \frac{m-k}{2}\right) := \frac{\int_0^{L_k^2/u} t^{\frac{k}{2}-1}(1-t)^{\frac{m-k}{2}-1} dt}{B(k/2, (m-k)/2)},$$

and the other case is similar. Here, for $u \le L_k^2$, we regard the probability is one. The other factor $\Pr[\|f\|^2 = u]$ can be computed by Claim 1. Therefore, the tail probability (14) can be computed efficiently by summing up $2(R_m^2 - L_m^2)$ terms.

We set L_k^2 and R_k^2 so that the both factors in the right-hand side of (14) are $p'/2(m-1)$ where $p' = \frac{mp-1}{m-1}$. Thus, we have $\Pr[\bar{E}_k|E_m] = p'/(m-1)$ for $k \in [m-1]$, and

$$\Pr[E_1 \cdots E_{m-1}|E_m] > 1 + \sum_{k=1}^{m-1} \Pr[E_k|E_m] = 1 - p'. \tag{15}$$

Therefore, with these settings, $p_{succ} > (1 - 1/m)(1 - p') = 1 - p$ holds. □

Practical relation between p' and p_{succ}: We remark on the success probability analysis. Since the inequality (15) uses a simple union bound, there exists a significant gap. To check it, we performed the preliminary experiment. Figure 2 shows the relation among dimension m, individual probability p', and the success probability p_{succ} when setting bounding function L_k and R_k so that $\Pr[E_k|E_m] = p'$ when $s = 8$. The circles and curves indicate the experimental result using the above formula, and the result of curve fitting where we set

$$p_{succ} = p^Y \text{ where } Y = a + bm^c + dp^e.$$

and

$$(a, b, c, d, e) = (0.0345, 3.32, 0.2, 7.93, 35.8).$$

Using this relation, we can set p when the target probability p_{succ} is given. We will use this formula to set p from a target $p_{succ} \in [0.01, 0.99]$.

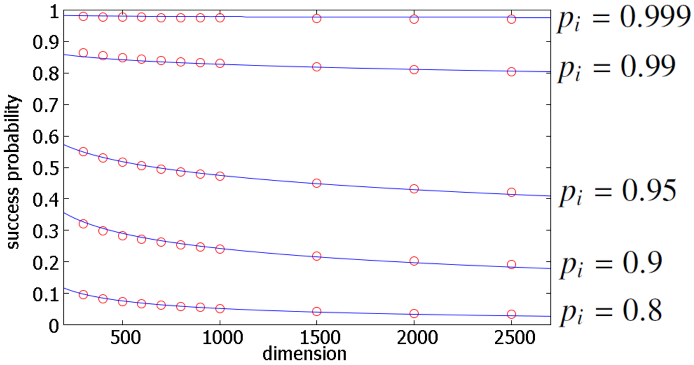

Figure 3: Relation between p and p_{succ} for several dimensions (temporal)

6 LWE Parameters

We give our experimental results of LWE attacking cost.

6.1 Comparison to Previous Results under GSA

Table 2 gives the comparison among previous and our attacks for Lindner-Peikert's parameters under GSA. We again remark that the GSA does not hold exactly and its estimation for timing of lattice vector enumeration small.

6.2 Updated Parameters

Table 3 gives updated attacking costs in various success probabilities from several published papers.

For given parameter set (n, q, s), we consider the additional parameters $(m, \beta, \sharp\text{Tours})$, and simulate $\|\widetilde{b_i}\|$ of m-dimensional q-ary lattices after ♯Tours tours of BKZ-β by using Chen-Nguyen's simulator. Then, the total attacking cost is given by the sum of (7) and enumeration cost in Section 4:

$$\text{Our Estimation}(n, q, s)[sec] = \text{Time}_{\text{BKZ}} + \sharp ENUM/(5 \cdot 10^7) \quad (16)$$

			Lindner-Peikert [17]		Liu-Nguyen [18]		**this work** with GSA (Our model 1)	
Gaussian model			Continuous		Continuous		Discrete	
$\log_2(t_{BKZ}(\delta))$			$1.8/\log_2(\delta) - 110$		[12]'s upper		$1.8/\log_2(\delta) - 130$	
$\sharp ENUM$/sec./thread			2^{15}		$10^7 = 2^{23.25}$ [12]		$5 \cdot 10^7 = 2^{25}$	
n	s	q	$\log_2$(**time in second in single thread**) and **success probability**					
128	6.77	2053	32	$\approx 100\%$	23.6	$\approx 63.21\%$	11.5	$\approx 95.4\%$
192	8.87	4093	78	$\approx 100\%$	62.8	$\approx 63.21\%$	52.4	$\approx 95.7\%$
256	8.35	4093	132	$\approx 100\%$	105.5	$\approx 63.21\%$	95.8	$\approx 95.7\%$
320	8.00	4093	189	$\approx 100\%$	–	–	139.7	$\approx 95.6\%$

Table 2: Comparison among several attacks on LWE using single-thread time for Lindner-Peikert parameters.All models assume that $\|\widetilde{b_i}\|$ of reduced lattices satisfy the GSA.

We search the optimal parameters ($m, \beta, \sharp$Tours) to minimize it by the standard numerical method.

Reference	Parameters			$\log_2$(our. est. [sec])		
	n	s	q	≈ 95 %	≈ 65 %	≈ 1 %
Lindner- Peikert [17]	128	6.77	2053	22.1	19	15
	192	8.87	4093	64.5	61.3	56.0
	256	8.35	4093	118	114	107
	320	8.00	4093	180	175	166
Aono et al. [6]	450	5.00	16381	186	182	175
	450	3.00	16381	148	144	138
RFID low [29]	152	7.292	8219	17.5	15.6	12.4
RFID high [29]	198	4.338	6803	31.4	29.2	25.7
Micciancio- Regev [20]	136	13.02	2003	53.8	50.1	44.3
	233	7.107	32749	38.8	36.7	33.4
LL15 [16]	350	8.0	2^{52}	< 0	< 0	< 0
	1024	8.0	$2^{47.5}$	58.1	57.5	56.0

Table 3: Updated attacking costs for previous parameters assuming Our model 2.

6.3 The Very Low Probability Case

In an application that involves data of a large number of persons, for example, 10^8 is the population of Japan, a data owner need to manipulate ciphertexts $Enc(pk_i, data_i)$ encrypted by many different secret keys. In such case, an attack with a low probability can be a threat to the system. We estimate Lindner-Peikert parameters by our attack using $p_{\text{succ}} \approx 10^{-8}$. The result is given in Table 4. Because the setting method of bounding functions are not investigated enough, the speed up is minor compared to the probability.

(n, q, s)	$\approx 95\%$	$\approx 10^{-8}$
$(192, 8.87, 4093)$	64.5	54.9
$(256, 8.35, 4093)$	118	101
$(320, 8.00, 4093)$	180	159

Table 4: Cost comparison between high and very low probabilities

6.4 An Explicit Formula for Parameter Setting

Since our prediction requires a heavy computation to obtain results, we give a formula of attacking time for parameter (n, s, q) by curve fitting. We assume the form of formula as

$$\log_2(Time_{LWE}\ [\text{sec}]) = \frac{Bn - C}{\ln q - A \ln s} - D.$$

which was derived from the estimation in [14]. Using the estimation in [17], they proposed a necessary LWE dimension $n \geq \log(q/s) \cdot (k + 110)/7.2$ to achieve 2^k attacking time/probability. Remark that the formula of [14] is a theoretical estimation. In contrast, our formula is from real experiments. In addition, the models under the formulas are also different, so it is hard to directly compare them.

With our hardness estimations for several parameters satisfies $n \in [100, 1000]$, $q \in [2^{10}, 2^{52}]$, $s \in [2.0, 14.0]$ and $p_{\text{succ}} \in [0.01, 0.95]$, besides Table 3, we fix the coefficients as (A, B, C, D) is $(1.66, 7.18, 219, 110)$ by the least square estimation. Figure 4 is the points whose coordinates are the security estimations by this formula and our method by (2). We can see the points are on the line $y = x$ which means our formula works well.

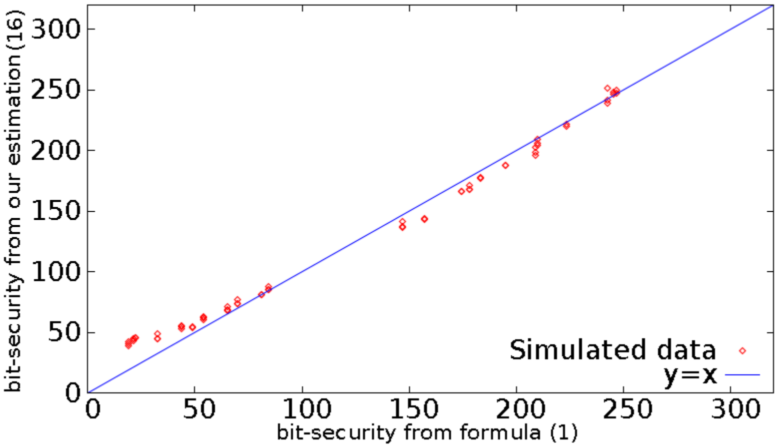

Figure 4: Verifying our security estimation formula

If one wants to obtain **bit security**, change D to 87. Furthermore, to obtain the lower bound of the attacking time of parameters, adjust D to 125.

Note that this estimation assumes $s > 1.5$ because if the deviation is too small, the lattice point enumeration can be done significantly fast. We also assume $p \geq 0.01$ because the estimated security (2) is much smaller than that by the formula when the probability is small. For low probability situation, it needs to perform the individual simulations.

7 Conclusion

We update the lattice based attack for LWE problem and adopt the previous attack to the discrete Gaussian model. We algo give an explicit formula to **bit security** estimation. Using this result, we can set the concrete parameters having any **bit security** in LWE based scheme.

References

[1] http://www.boost.org/, Title = Boost C++ Libraries.

[2] TU Darmstadt SVP Challenge. http://www.latticechallenge.org/svp-challenge/.

[3] M. R. Albrecht, C. Cid, J. Faugère, R. Fitzpatrick, and L. Perret. On the complexity of the BKW algorithm on LWE. *Des. Codes Cryptography*, 74(2):325–354, 2015.

[4] M. R. Albrecht, C. Cid, J.-C. Faugere, R. Fitzpatrick, and L. Perret. Algebraic algorithms for LWE problems. Cryptology ePrint Archive, Report 2014/1018, 2014. http://eprint.iacr.org/.

[5] M. R. Albrecht, R. Player, and S. Scott. On the concrete hardness of learning with errors. Cryptology ePrint Archive, Report 2015/046, 2015. http://eprint.iacr.org/.

[6] Y. Aono, X. Boyen, L. T. Phong, and L. Wang. Key-private proxy re-encryption under LWE. In G. Paul and S. Vaudenay, editors, *INDOCRYPT*, volume 8250 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2013.

[7] S. Arora and R. Ge. New algorithms for learning in presence of errors. In L. Aceto, M. Henzinger, and J. Sgall, editors, *ICALP (1)*, volume 6755 of *Lecture Notes in Computer Science*, pages 403–415. Springer, 2011.

[8] S. Bai and S. D. Galbraith. Lattice decoding attacks on binary LWE. In *ACISP*, pages 322–337, 2014.

[9] A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, July 2003.

[10] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. Classical hardness of learning with errors. In D. Boneh, T. Roughgarden, and J. Feigenbaum, editors, *STOC*, pages 575–584. ACM, 2013.

[11] J. Buchmann and C. Ludwig. Practical lattice basis sampling reduction. In F. Hess, S. Pauli, and M. Pohst, editors, *Algorithmic Number Theory*, volume 4076 of *Lecture Notes in Computer Science*, pages 222–237. Springer Berlin Heidelberg, 2006.

[12] Y. Chen and P. Q. Nguyen. BKZ 2.0: Better lattice security estimates. In D. H. Lee and X. Wang, editors, *ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2011.

[13] N. Gama, P. Q. Nguyen, and O. Regev. Lattice enumeration using extreme pruning. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 257–278. Springer, 2010.

[14] C. Gentry, S. Halevi, and N. P. Smart. Homomorphic evaluation of the AES circuit. In *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, pages 850–867, 2012.

[15] R. Kannan. Improved algorithms for integer programming and related lattice problems. In D. S. Johnson, R. Fagin, M. L. Fredman, D. Harel, R. M. Karp, N. A. Lynch, C. H. Papadimitriou, R. L. Rivest, W. L. Ruzzo, and J. I. Seiferas, editors, *STOC*, pages 193–206. ACM, 1983.

[16] K. Laine and K. Lauterr. Key recovery for lwe in polynomial time. Cryptology ePrint Archive, Report 2015/176, 2015. `http://eprint.iacr.org/`.

[17] R. Lindner and C. Peikert. Better key sizes (and attacks) for LWE-based encryption. In A. Kiayias, editor, *CT-RSA*, volume 6558 of *Lecture Notes in Computer Science*, pages 319–339. Springer, 2011.

[18] M. Liu and P. Q. Nguyen. Solving BDD by enumeration: An update. In E. Dawson, editor, *CT-RSA*, volume 7779 of *Lecture Notes in Computer Science*, pages 293–309. Springer, 2013.

[19] S. Liu, C. Ling, and D. Stehlé. Decoding by sampling: A randomized lattice algorithm for bounded distance decoding. *IEEE Transactions on Information Theory*, 57(9):5933–5945, 2011.

[20] D. Micciancio and O. Regev. Lattice-based cryptography. In *Post-Quantum Cryptography*, pages 147–191. Springer, 2009.

[21] P. Q. Nguyen and D. Stehlé. LLL on the average. In *Algorithmic Number Theory, 7th International Symposium, ANTS-VII, Berlin, Germany, July 23-28, 2006, Proceedings*, pages 238–256, 2006.

[22] P. Q. Nguyen and B. Valle. *The LLL Algorithm: Survey and Applications*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[23] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In H. N. Gabow and R. Fagin, editors, *STOC*, pages 84–93. ACM, 2005.

[24] C.-P. Schnorr. Lattice reduction by random sampling and birthday methods. In H. Alt and M. Habib, editors, *STACS 2003*, volume 2607 of *Lecture Notes in Computer Science*, pages 145–156. Springer, 2003.

[25] C. P. Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. In *Math. Programming*, pages 181–191, 1993.

[26] B. Shim and I. Kang. Sphere decoding with a probabilistic tree pruning. *IEEE Transactions on Signal Processing*, 56(10-1):4867–4878, 2008.

[27] V. Shoup. A library for doing Number Theory. `www.shoup.net/ntl`.

[28] R. Smith. Efficient Monte-Carlo procedures for generating points uniformly distributed over bounded regions. *Operations Res.*, 32:1296–1308, 1984.

[29] Y. Yao, J. Huang, S. Khanna, A. Shelat, B. H. Calhoun, J. Lach, and D. Evans. A sub-0.5V lattice-based public-key encryption scheme for RFID platforms in 130nm CMOS.

A Random Sampling Method to Approximate the Volume of $C(l_k, r_k)$ in the Conference Version [6]

We also define the covering object for an even integer $k \geq 2$ by

$$C'(l_k, r_k) = \left\{ (x_1, \dots, x_k) \in \mathbb{R}^k : L_i^2 < \sum_{\ell=1}^{i} x_\ell^2 < R_i^2 \text{ for } \forall \text{ even } i \in [k] \right\}.$$

Clearly, $C(l_k, r_k) \subset C'(l_k, r_k)$.

An algorithm for approximating $\mathrm{Vol}C(l_k; r_k)$ by induction is given as follows. For the base cases $k = 1$ and 2, they can be computed by $\mathrm{Vol}C(L_1; R_1) = 2(R_1 - L_1)$ and approximated by the standard Monte-Carlo sampling method, respectively. For simplicity, let us denote the interval $I_k = [-R_k, R_k]$.

For even $k \geq 2$, suppose (an approximation of) $\mathrm{Vol}C(l_k; r_k)$ is computed. Then by the relation in $(k+1)$-dimensional space

$$C(r_{k+1}, l_{k+1}) \subset C(r_k, l_k) \times [-R_{k+1}, R_{k+1}] \subset C'(r_k, l_k) \times [-R_{k+1}, R_{k+1}],$$

the volume in $k + 1$ dimension is computed by the relation

$$\mathrm{Vol}C(r_{k+1}, l_{k+1}) = \mathrm{Vol}C(r_k, l_k) \cdot 2R_{k+1} \times \Pr\left[x \in C(r_{k+1}, l_{k+1}) \middle| x \in C(r_k, l_k) \times I_{k+1} \right].$$

where the probability is over $x \leftarrow C'(r_k, l_k) \times [-R_{k+1}, R_{k+1}]$. Here, the above holds for any probability distribution over $x \leftarrow S$ such that $S \supset x \in C(r_k, l_k) \times [-R_{k+1}, R_{k+1}]$. Following [13], we decided to take $S = C'(r_k, l_k) \times [-R_{k+1}, R_{k+1}]$ that gives a better balance between the easiness of sampling and probability ratio $\Pr[x \in C(r_k, l_k) \times [-R_{k+1}, R_{k+1}]]$. As shown below, uniform sampling from $C'(r_k, l_k)$ is easy and hence the probability is easily approximated.

Next, consider the $(k+2)$-dimensional object. By the relation

$$C(r_{k+2}, l_{k+2}) \subset C(r_k, l_k) \times I_{k+1} \times I_{k+2} \subset C'(r_k, l_k) \times I_{k+1} \times I_{k+2},$$

we can also compute the volume by

$$\mathrm{Vol}C(r_{k+2}, l_{k+2}) = \mathrm{Vol}C(r_k, l_k) \cdot 4R_{k+1}R_{k+2} \times \Pr\left[x \in C(r_{k+2}, l_{k+2}) \middle| x \in C(r_k, l_k) \times I_{k+1} \times I_{k+2} \right]$$

where the probability is over $x \leftarrow C'(r_k, l_k) \times I_{k+1} \times I_{k+2}$.

Uniform sampling from even-tube-intersections. To approximate the probability, we need to perform uniform sampling from $C'(r_k, l_k)$. This can be done by generating

$$(\sqrt{u_1}\cos\theta_1,\ \sqrt{u_1}\sin\theta_1,\ \sqrt{u_2}\cos\theta_2, \dots,\ \sqrt{u_{k/2}}\cos\theta_{k/2}) \in \mathbb{R}^k \tag{17}$$

where $(\theta_1, \dots, \theta_{k/2})$ is uniform over $[-\pi, \pi]^{k/2}$ and $u = (u_1, \dots, u_{k/2})$ is uniform from the polygon

$$P(r_k, l_k) := \left\{ u \in \mathbb{R}^{k/2} \middle| L_{2i}^2 < \sum_{\ell=1}^{i} u_\ell < R_{2i}^2,\ u_i \in [0, 1]\ \forall i \in \left[\frac{k}{2} \right] \right\}$$

by the hit-and-run algorithm [28] for sampling points uniformly from this object.

The correctness follows from [13]. Here we give a proof outline. Let us consider the standard $(k/2)$-simplex

$$\Delta_{k/2} = \left\{ w \in \mathbb{R}^{k/2} \middle| w_i \geq 0 \text{ for } \forall i \in [k/2] \text{ and } \sum_{\ell=1}^{k/2} w_\ell \leq 1 \right\}$$

that contains $C'(r_k, l_k)$, and let $(y_1, \ldots, y_{k/2})$ be the uniform sampling from the simplex. Then, the extended vector $(y_1, \ldots, y_{k/2}, \bar{y})$ where $\bar{y} = 1 - y_1 - \cdots - y_{k/2}$ has the Dirichlet distribution $Dir(1, \ldots, 1)$ of order $k/2 + 1$. On the other hand, for θ over uniform $[-\pi, \pi]$, $(\cos^2\theta, \sin^2\theta)$ has the distribution $Dir(1/2, 1/2)$. Hence, the compound random distribution is a tuple in $\mathbb{R}^{k+2}$

$$(y_1 \cos^2\theta_1, y_1 \sin^2\theta_1, y_2 \cos^2\theta_2, \ldots, y_{k/2} \cos^2\theta_{k/2}, \bar{y}\cos^2\theta_{k/2+1}, \bar{y}\cos^2\theta_{k/2+1})$$

where θ_i are uniformly sampled from $[-\pi, \pi]$ independently, has the distribution $Dir(1/2, \ldots, 1/2)$. Thus, by a straightforward computation of probability density function, the component-wise squared distribution

$$(\sqrt{y_1}\,|\cos\theta_1|, \sqrt{y_1}\,|\sin\theta_1|, \ldots, \sqrt{y_{k/2}}\,\left|\sin\theta_{k/2}\right|, \sqrt{\bar{y}}\,\left|\sin\theta_{k/2+1}\right|, \sqrt{\bar{y}}\,\left|\cos\theta_{k/2+1}\right|) \in \mathbb{R}^{k+2}$$

is the uniformly random over the part of $(k+2)$-dimensional unit sphere

$$\left\{ (z_1, \ldots, z_{k+2}) \in \mathbb{R}^{k+2} : z_i \geq 0 \text{ for } \forall\ i \text{ and } \sum_{i=1}^{k+2} z_i^2 = 1) \right\}.$$

Therefore, extracting first k coordinates and removing the absolute function, it can be shown that

$$(\sqrt{y_1}\cos\theta_1, \sqrt{y_1}\sin\theta_1, \ldots, \sqrt{y_{k/2}}\cos\theta_{k/2}) \in \mathbb{R}^k$$

is the uniform distribution in the unit ball.

Finally, considering rejection sampling, that is, restricting $\Delta_{k/2}$ to $P(r_k, l_k)$ corresponds to restricting the unit ball to $C'(r_k, l_k)$.