Journal of Information Processing Vol.20 No.3 785–795 (July 2012)

[DOI: 10.2197/ipsjjip.20.785]

Regular Paper

Finding a Very Short Lattice Vector in the Extended Search Space

Masaharu Fukase[1,a)] Kazunori Yamaguchi[2,b)]

Received: October 5, 2011, Accepted: April 2, 2012

Abstract: The problem of finding a lattice vector approximating a shortest nonzero lattice vector (approximate SVP) is a serious problem that concerns lattices. Finding a lattice vector of the secret key of some lattice-based cryptosystems is equivalent to solving some hard approximate SVP. We call such vectors very short vectors (VSVs). Lattice basis reduction is the main tool for finding VSVs. However, the main lattice basis reduction algorithms cannot find VSVs in lattices in dimensions ∼200 or above. Exhaustive search can be considered to be a key technique toward eliminating the limitations with current lattice basis reduction algorithms. However, known methods of carrying out exhaustive searches can only work in relatively low-dimensional lattices. We defined the extended search space (ESS) and experimentally confirmed that exhaustive searches in ESS make it possible to find VSVs in lattices in dimensions ∼200 or above with the parameters computed from known VSVs. This paper presents an extension of our earlier work. We demonstrate the practical effectiveness of our technique by presenting a method of choosing the parameters without known VSVs. We also demonstrate the effectiveness of distributed searches.

Keywords: lattice, approximate SVP, exhaustive search, enumeration

1. Introduction

An integer lattice, L, is the set of all linear combinations with integer coefficients of a set of linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{Z}^m$. The problem of finding a nonzero lattice vector with the shortest length for a given lattice basis has been one of the most widely studied problems concerning lattices. Note that such vectors are not necessarily unique. This problem is called the shortest vector problem (SVP). No polynomial time algorithm for SVP is known for lattices with arbitrary dimensions, and Ajtai proved SVP to be NP-hard under randomized reduction [2]. It is already difficult to determine if a given short lattice vector is a shortest nonzero lattice vector or not. One can at most expect that a given short lattice vector will likely be a shortest nonzero lattice vector with a heuristic threshold for random lattices. Micciancio [14] stated that the lack of efficient algorithms for SVP led compute r scientists to consider approximation versions of it. The approximation problem for SVP is to find a nonzero lattice vector with at most γ-times the minimal possible length for constant γ. This approximation problem is called approximate SVP, and a solution to it is called an approximate shortest vector. When γ is small, approximate SVP is still hard. In fact, Micciancio proved approximate SVP to be NP-hard for $\gamma < \sqrt{2}$ [15]. In this and our earlier research, we constructed an exhaustive search to solve approximate SVP.

Many lattice basis reduction algorithms for approximate SVP have been proposed in the literature, e.g., the Lenstra, Lenstra, and Lovász (LLL) [12], the block Korkine-Zolotarev (BKZ) [17], and the random sampling reduction (RSR) [18] algorithms. They are key tools for attacking lattice-based cryptosystems [1], [8], [10], [14]. The secret key in most lattice-based cryptosystems is a short vector or short vectors in a particular class of lattices. A lattice vector of the secret key is not only short but it also approximates a shortest nonzero lattice vector within a small factor, γ. Consequently, finding a lattice vector of the secret key is a difficult task because this is equivalent to solving some hard approximate SVP. However, lattice basis reduction algorithms occasionally recover the secret key of lattice-based cryptosystems for a given public key in lattices in dimensions above 100. For example, Sc hnorr recovered the secret key by using RSR given the corresponding public key of GGH cryptosystem in lattices in dimension 180 [18], and Gama and Nguyen broke NTRU cryptosystems in lattices in dimension 214 by using BKZ and some improved reduction [6]. These examples also reveal the limitations with current lattice basis reduction algorithms. Exhaustive searches can be considered to be a key technique to eliminating these limitations. However, Schnorr-Euchner's enumeration [17], which was the most efficient known method for exhaustive searches before Schnorr [18] and Gama et al. [7], could only work in lattices in dimensions below 100. Fukase et al. [5] improved the shape of the search space used by Schnorr [18] so that it included a shortest nonzero lattice vector with higher probability.

We must determine if a found lattice vector is a shortest nonzero lattice vector or not to evaluate how well our technique performs. Although this is generally difficult, a shortest nonzero lattice vector is heuristically known in some cryptographic applications. For example, a lattice vector of the secret key in the GGH cryptosystem has been treated as a shortest nonzero lattice

[1] Dokkyo University, Souka, Saitama 340–0042, Japan
[2] The University of Tokyo, Meguro, Tokyo 153–8902, Japan
[a)] fukase@dokkyo.ac.jp
[b)] yamaguch@graco.c.u-tokyo.ac.jp

vector [18]. Hoffstein et al. [10] showed that the secret key of the NTRU cryptosystem was heuristically a shortest nonzero lattice vector, and Gama et al. [6] stated that the secret key of the NTRU cryptosystem was related to linearly independent shortest vectors. We followed these examples and treated a lattice vector of the secret key in some lattice-based cryptosystems as a shortest nonzero lattice vector. However, to retain rigor in the definition of a shortest nonzero lattice vector, we made a distinction between a lattice vector of the secret key and a sho rtest nonzero lattice vector. Ludwig [13] used the term "a very short vector" to represent a lattice vector of the secret key of an NTRU cryptosystem. A very short vector (VSV) means a lattice vector of the secret key of some lattice-based cryptosystems in the rest of this paper.

This work is a continuation of that by Fukase et al. [4], [5]. We defined the extended search space (ESS) and experimentally confirmed that the exhaustive search in ESS made it possible to find a VSV in lattices in dimensions ∼200 or above with the parameters computed from known VSVs [5]. This research extends our earlier work so that appropriate parameters for ESS can be determined without known VSVs in practical situations where VSVs are not known. We mean parameters that maximize the inclusion probability of a VSV through appropriate parameters. We utilize the probabilistic distribution of the probabilistic variable related to the Gram-Schmidt coefficients of VSVs to compute the inclusion probability of a VSV, which can be obtained in some cryptographic situations.

We also report experimental results obtained from a distributed search in ESS. It has been pointed out that the search of a lattice vector using a sampling algorithm [18] or its variants can easily be distributed [13]. This is also the case for the search in ESS. We experimentally demonstrated the distributed search in ESS was effective, which is discussed in this paper.

This paper makes two main contributions.

(1) We constructed a method of computing the inclusion probability of a VSV in ESS and outputting the refined parameters for ESS to maximize the inclusion probability. We introduced the distribution of a deviation from the estimated values of coefficients of VSVs to compute the inclusion probability. We found that the parameters computed with the method were sufficient to achieve high inclusion ratios for VSVs.

(2) We experimentally confirmed the distributed search in ESS was effective. Significant speedups were achieved on eight CPUs in some small-scale experiments.

The remainder of this paper is organized as follows. Section 2 explains some basic concepts of lattices. We recall the definition of ESS and explain the performance of the search in ESS with the parameters computed from known VSVs in Section 3. In Section 4, we present a scheme to compute the refined parameters for the ESS of a given basis. In Section 5, we report experimental results on the distributed search in ESS. Section 6 concludes the paper.

Related Work

Extreme pruning [7] has recently been proposed for lattice enumeration. Its analysis is based on the distribution of a shortest nonzero lattice vector, like that with our method. It has achieved exponential speedups for Schnorr-Euchner's enumeration. There are some similarities between our method [4], [5] and Gama et al. [7], but the bounding functions and types of lattices that are targeted are very different. Also, the strategy for search is very different. Gama et al. [7] search space size was drastically reduced by pruning the search tree and the searches were conducted many times to compensate for the loss in the success probability caused by pruning the search tree. Our strategy for search, on the other hand, was to reduce the size of the search space sufficiently to enable an efficient search, to simultaneously make the success probability as high as possible, and to basically conduct the search once. Therefore , it seems that each method has different advantages and should be evaluated from slightly different aspects. After Gama et al. [7], the effectiveness of parallelizing enumeration with multi-core CPUs, GPUs, or cloud-computing was reported [3], [9], [11]. This paper reports parallelization was effective with our search method of only using multi-core CPUs. We also intend to parallelize our search method with GPUs or cloud-computing in future work.

2. Preliminaries

2.1 Lattice

Given a set of n linearly independent vectors $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{Z}^{m\times n}$, the *integer lattice* $L \subset \mathbb{Z}^m$ spanned by B is defined as the set, $L(B) = \{B\mathbf{x} \mid \mathbf{x} \in \mathbb{Z}^n\}$, of all integral combinations of $\mathbf{b}_i$'s. The integer, n, is called the *dimension* of L. When $n = m$, we say that L is *full-dimensional*. The ordered set of vectors $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{Z}^{m\times n}$ is called a *basis* of L. We concentrate on full-dimensional integer lattices in this paper. A lattice has infinitely many bases that generate a lattice when $n \geq 2$. For lattice basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$, the corresponding *Gram-Schmidt orthogonalized vectors*, $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^* \in \mathbb{R}^n$, are defined by $\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j}\mathbf{b}_j^*$ with $\mu_{i,j} = \langle \mathbf{b}_i, \mathbf{b}_j^* \rangle / \langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle$ where $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{n} x_i y_i$ is the inner product in $\mathbb{R}^n$. For every i, $\mathbf{b}_i^*$ is the component of $\mathbf{b}_i$ that is orthogonal to $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$. Consequently, vectors $\mathbf{b}_i^*$ and $\mathbf{b}_j^*$ $(j \neq i)$ are orthogonal.

Let $\mathbf{v} = B\mathbf{x}$ with $\mathbf{x} \in \mathbb{Z}^n$ be a vector in the lattice generated by the basis, B. From the definition of Gram-Schmidt orthogonalized vectors, we can represent $\mathbf{v}$ with $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$ and the $\mu_{i,j}$ of B, i.e., $\mathbf{v} = \sum_{j=1}^{n} \nu_j \mathbf{b}_j^*$ with $\boldsymbol{\nu} \in \mathbb{R}^n$ such that $\nu_j = \sum_{i=1}^{n} x_i \mu_{i,j}$. Because $\mathbf{b}_j^*$ are pairwise orthogonal, $\|\mathbf{v}\|^2 = \sum_{j=1}^{n} \nu_j^2 \|\mathbf{b}_j^*\|^2$. This equation means that for lattice vector $\mathbf{v} = \sum_{j=1}^{n} \nu_j \mathbf{b}_j^*$ to be short $|\nu_j|$ for $j = 1, \ldots, n$ needs to be small. In the following, we call ν_j *the Gram-Schmidt coefficients* of $\mathbf{v}$.

We denote the length of the shortest nonzero lattice vector in lattice L by $\lambda_1(L)$ or λ_1 when L is considered to be obvious.

2.2 Lattice Basis Reduction Algorithms

Lattice basis reduction algorithms are key tools for approximate SVP. Several different lattice basis reduction algorithms have been proposed in the literature. The BKZ algorithm [17] computes a (δ, β)-BKZ reduced basis for $\delta \in (1/4, 1]$ and an integer β such that $2 \leq \beta < n$. There is no proven polynomial time bound for the BKZ algorithm, but it behaves well for reasonable β. Although the quality of a (δ, β)-BKZ reduced basis is better for larger δ and β, the computational cost increases for larger δ and

β.

Schnorr proposed random sampling reduction (RSR) [18], which is applied to lattice bases whose Gram-Schmidt orthogonalized basis satisfies *the geometric series assumption* (*GSA*). GSA states that for lattice basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$, there is $q \in [0, 1]$ such that $\|\mathbf{b}_j^*\|^2 = q^{j-1}\|\mathbf{b}_1\|^2$ for $j = 1, \ldots, n$. We call q the common ratio of B. Quotients $\|\mathbf{b}_j^*\|^2/\|\mathbf{b}_1\|^2$ of basis B just approximate q^{j-1} in practice. It is well known that the initial vectors, $\mathbf{b}_1^*, \ldots, \mathbf{b}_k^*$, for some $1 \le k < n$ are longer than subsequent vectors $\mathbf{b}_j^*$ for $j > k$ if B is reduced by BKZ. Consequently, Gram-Schmidt coefficients $\nu_1, \ldots, \nu_k$ have a larger impact on the overall length of $\mathbf{v}$ than ν_j for $j > k$. Recall that for lattice vector $\mathbf{v} = \sum_{j=1}^{n} \nu_j \mathbf{b}_j^*$ to be short, $|\nu_j|$ needs to be small. Then, it is reasonable to assume that vector $\mathbf{v} = \sum_{j=1}^{n} \nu_j \mathbf{b}_j^*$ such that

$$|\nu_j| \le \begin{cases} \frac{1}{2} & \text{for } j < n-u \\ 1 & \text{for } n-u \le j < n \end{cases}, \quad \nu_n = 1 \tag{1}$$

for some $1 \le u \le n$ is likely to be short. There are 2^u distinct lattice vectors with this form. The sampling algorithm (SA) generates a single vector, $\mathbf{v}$, satisfying Eq. (1). Let $S_{u,B}$ be the set of lattice vectors in $L(B)$ satisfying Eq. (1) for the specified u. We call $S_{u,B}$, the *SA search space*.

Sampling Algorithm (SA)

Input:

lattice basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ with $\mu_{i,j}$ and an integer, u, such that $1 \le u < n$.

Output:

$\mathbf{v}$ satisfying Eq. (1).

Procedure:

$\mathbf{v} := \mathbf{b}_n$

for $j = 1, \ldots, n-1$ $\mu_j := \mu_{n,j}$

for $i = n-1, \ldots, 1$

select $y \in \mathbb{Z}$ randomly such that $|\mu_i - y| \le \begin{cases} 1/2 & \text{if } i < n-u \\ 1 & \text{if } i \ge n-u \end{cases}$

$\mathbf{v} := \mathbf{v} - y\mathbf{b}_i$

for $j = 1, \ldots, n-1$ $\mu_j := \mu_j - y\mu_{i,j}$

Given lattice basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$, RSR samples by calling SA up to 2^u distinct lattice vectors $\mathbf{v} = \sum_{j=1}^{n} \nu_j \mathbf{b}_j^*$ satisfying Eq. (1) until a vector, $\mathbf{v}$, such that $\|\mathbf{v}\|^2 < 0.99\|\mathbf{b}_1\|^2$ is found. RSR subsequently inserts the vector found by SA into the basis, and BKZ is used to reduce the new basis, $\mathbf{v}, \mathbf{b}_1, \ldots, \mathbf{b}_n$. This random sampling by SA and the BKZ process are iterated several times.

2.3 Lattice Based Cryptosystems

We applied our method to three types of lattices that were related to lattice-based cryptosystems in this study.

2.3.1 GGH Cryptosystem

The idea of public key construction in the GGH cryptosystem is that it is hard to find a basis consisting of considerably short vectors from a basis consisting of very long vectors in a lattice. Based on this idea, the private key of the GGH cryptosystem is set to the former and the public key is set to the latter. The private basis, R, is defined as $R = kI + R'$ with $k \in \mathbb{Z}$. R' is a perturbation matrix with entries chosen independently and uniformly at random from $\{-l, \ldots, +l\}$, and I is the identity matrix. R is transformed into a public basis, B, by applying elementary column operations $2n$ times. We add a random integer combination of the other columns to a column at every step. The coefficients in the integer combination are chosen at random from $\{-1, 0, +1\}$. We call the lattices generated by private bases or public bases in GGH cryptosystems *GGH lattices*.

2.3.2 Micciancio's GGH Cryptosystem

Micciancio [14] improved the GGH cryptosystem with the HNF technique, where the key and ciphertext sizes were reduced by a factor, n, without decreasing security. Consider a lattice basis, K, whose matrix is uniformly chosen in $\{-n, \ldots, n\}^{n \times n}$. Micciancio proposed a private basis, R, to be an LLL reduced basis of K. The corresponding public basis is the Hermite normal form of R. A lattice basis, $H = (h_{i,j}) \in \mathbb{Z}^{n \times n}$, is said to be in the *Hermite normal form* if and only if H is upper triangular and $0 \le h_{i,j} < h_{i,i}$ for all $1 \le i < j \le n$. Every lattice has exactly one basis H in the Hermite normal form. We call the lattices generated by private bases or public bases in Micciancio's GGH cryptosystems *Micciancio's GGH lattices*. While the private basis is cube-like in the GGH cryptosystem, the private basis is an LLL-reduced basis of a basis whose vectors are chosen at random in an n-dimensional c ube in Micciancio's GGH cryptosystem. Thus, compared with GGH lattices, Micciancio's GGH lattices do not have any particular structure. Therefore, Micciancio's GGH lattices are very suitable for estimating the generality of our results.

2.3.3 NTRU Cryptosystem

NTRU lattices underlie the NTRU cryptosystem [10]. The key generation process and the suggested parameters of NTRU have been revised several times. For simplicity, we followed the original description of NTRU in Hoffstein et al. [10]. The private key, (f, g), and the public key, h, in NTRU are polynomials in polynomial rings, and h is constructed from (f, g). The encryption function and the decryption function are also based on arithmetic in polynomial rings. However, it was shown that breaking NTRU is related to finding short vectors in a particular class of lattices. In particular, private key (f, g) was shown to be heuristically equivalent to the shortest nonzero lattice vector in the class of lattices.

Private key (f, g) can be represented as a $2N$-dimensional vector, $(f, g) = [f_0, \ldots, f_{N-1}, g_0, \ldots, g_{N-1}]$, where N is the security parameter of NTRU. Public key h can be represented as an N-dimensional vector, $h = [h_0, \ldots, h_{N-1}] \in \mathbb{Z}^n$. Consider the following matrix, M_h with h:

$$M_h = \begin{bmatrix} h_0 & h_1 & \cdots & h_{N-1} \\ h_{N-1} & h_0 & \cdots & h_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ h_1 & h_2 & \cdots & h_0 \end{bmatrix} \in \mathbb{Z}^{N \times N}. \tag{2}$$

Next, consider matrix B below with N, an integer, q, which is a power of 2, and M_h:

$$B = \begin{bmatrix} I & \mathbf{0} \\ M_h & qI \end{bmatrix} \in \mathbb{Z}^{2N \times 2N}, \tag{3}$$

where I is the N-dimensional identity matrix. We call the lattices generated by bases represented by matrix B *NTRU lattices*. Matrix B above is the simplest form of bases for NTRU lattices. The

dimension of NTRU lattices is $2N$. Private key (f, g) is a lattice vector in the NTRU lattice generated by the corresponding public key, h.

Hoffstein et al. [10] presented three standard sets of parameters. We will concentrate on the first of $(N, q) = (107, 64)$ in this paper. The level of security yielded by this parameter set is lowest of the three, but no NTRU lattices of $N = 107$ had ever been broken only by lattice reduction before Gama et al. [6]. Although three NTRU lattices of $N = 107$ were broken by using BKZ and some improved reduction by Gama et al. [6], the NTRU lattices of $N = 107$ can be considered to still be hard for BKZ alone.

2.4 Very Short Vector (VSV)

We define a very short vector (VSV) as a lattice vector of the secret key of some lattice-based cryptosystems in this paper. We call $\|\mathbf{v}\|/\lambda_1(L)$ for lattice vector $\mathbf{v}$ an approximation factor following Gama et al. [6], and denote it as $apfa$ for short. Finding a VSV is equivalent to solving approximate SVP for small γ, which means a VSV is an approximate shortest vector with $apfa \leq \gamma$. We support this standpoint by referring to Gama et al. [6] where they explained that finding a lattice vector of the secret key of NTRU cryptosystems was equivalent to solving approximate SVP for a suitable γ. It is difficult to determine how small γ is because $\lambda_1(L)$ for a lattice, L, is generally not exactly known.

3. ESS

This section explains the definition of ESS and discusses the performance of search in ESS with the parameters computed from known VSVs.

3.1 Definition

We analyzed the distribution of coefficients of VSVs and demonstrated that it is related to some increasing geometric sequence [5]. Here, we recall the results. Let $R = [\mathbf{r}_1, \ldots, \mathbf{r}_n]$ be a GGH private basis and let $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ be a $(0.99, \beta)$-BKZ reduced GGH public basis. We represent each vector $\mathbf{r}_i$ of R with the Gram-Schmidt orthogonalized vectors $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$ as $\mathbf{r}_i = \sum_{j=1}^{n} \nu_{i,j}\mathbf{b}_j^*$. $\nu_{i,j}$ is computed as $\nu_{i,j} = \sum_{i=1}^{n} u_i\mu_{i,j}$ with the unimodular matrix, $U = [\mathbf{u}_1, \ldots, \mathbf{u}_n]$, such that $R = BU$. $\mu_{i,j}$ are the coefficients defined as $\mu_{i,j} = \langle \mathbf{b}_i, \mathbf{b}_j^* \rangle / \langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle$ in Section 2.1.

Now, we will investigate the distribution of the Gram-Schmidt coefficients, $\nu_{i,j}$.

Figure 1 plots the distribution of $|\nu_{i,j}|$ for the $(0.99, 10)$-BKZ reduced GGH basis in dimension 200. It can be seen that $|\nu_{i,j}|$ are related to some increasing geometric sequence. **Figure 2** shows the distribution of $|\nu_{i,j}|$ for the $(0.99, 20)$-BKZ reduced GGH basis in dimension 200. By comparing Fig. 1 with Fig. 2, we can see that $|\nu_{i,j}|$ for the $(0.99, 20)$-BKZ reduced basis is smaller than those for the $(0.99, 10)$-BKZ reduced basis. We tested this on many bases, and we witnessed the same tendency: $|\nu_{i,j}|$ is smaller for the better reduced basis.

From the above observation, we defined ESS $W_{k,a,j_0,B}$ as follows [5]. Here, we employ a geometric sequence, ka^{n-j}, to bound a search space.

Definition 1 Let B be a lattice basis, and let $k, a \in \mathbb{R}^+$, $j_0 \in \mathbb{Z}_n^+$. Then, ESS $W_{k,a,j_0,B}$ is the set of all lattice vectors

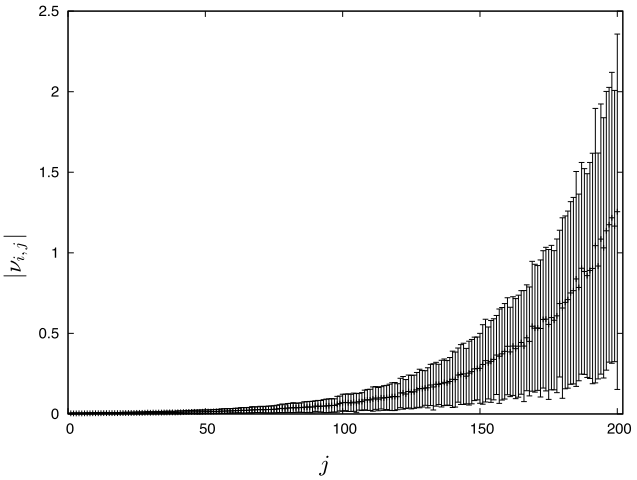

Fig. 1 $|\nu_{i,j}|$ for $\mathbf{r}_i = \sum_{j=1}^{n} \nu_{i,j}\mathbf{b}_j^*$. $|\nu_{i,j}|$ are averaged over i $(1 \leq i \leq n)$. Each error bar represents the standard deviation of the average. $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$ are the Gram-Schmidt vectors of a $(0.99, 10)$-BKZ reduced GGH basis in dimension 200. These results were presented in Fukase et al. [5].

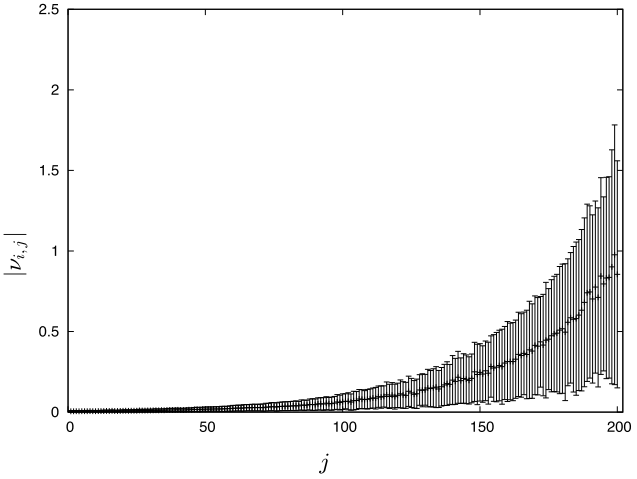

Fig. 2 $|\nu_{i,j}|$ for $\mathbf{r}_i = \sum_{j=1}^{n} \nu_{i,j}\mathbf{b}_j^*$. $|\nu_{i,j}|$ are averaged over i $(1 \leq i \leq n)$. Each error bar indicates the standard deviation of the average. $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$ are the Gram-Schmidt vectors of a $(0.99, 20)$-BKZ reduced GGH basis in dimension 200. These results were presented in Fukase et al. [5].

$\mathbf{v} = \sum_{j=1}^{n} \nu_j\mathbf{b}_j^*$ with $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_n) \in \mathbb{R}^n$ such that

$$\nu_j \in \begin{cases} (-\lceil 2ka^{n-j} \rceil/2, \lceil 2ka^{n-j} \rceil/2] & \text{for } 1 \leq j < j_0, \\ \{1, \ldots, \lceil ka^{n-j} \rfloor\} & \text{for } j = j_0, \\ \{0\} & \text{for } j_0 < j \leq n, \end{cases} \tag{4}$$

where $\lceil x \rfloor$ rounds x to the closest integer as defined by $\lceil x \rfloor = \lceil x - 0.5 \rceil$.

Because $S_{u,B} = W_{k,a,j_0,B}$ with $k = 1.0$, $a = (0.5)^{\frac{1}{u+1}}$, and $j_0 = n$, the SA search space is a special case of ESS.

We explain the search in ESS in the following. From Definition 1, it can be seen that the forms for the upper bounds of $|\nu_j|$ of a vector in ESS are $z/2$ with some $z \in \mathbb{N}$. This enables us to efficiently enumerate ESS with GenSample [13], and in particular $|W_{k,a,j_0,B}| = \prod_{j=1}^{j_0} c_j$ with $\mathbf{c} = (c_1, \ldots, c_n) \in \mathbb{N}^n$ such that

$$c_j = \begin{cases} \lceil 2ka^{n-j} \rceil & \text{if } 1 \leq j < j_0, \\ \lceil ka^{n-j} \rfloor & \text{if } j = j_0, \\ 0 & \text{if } j_0 < j \leq n, \end{cases} \tag{5}$$

for $j = 1, \ldots, n$. GenSample can be used for the search in $W_{k,a,j_0,B}$. GenSample takes B with $\mu_{i,j}$, some $\mathbf{c}' = (c'_1, \ldots, c'_n) \in \mathbb{N}^n$ and some $x \in \mathbb{N}$ as input and outputs a lattice vector in the set specified by $\mathbf{c}'$. Here, $\mu_{i,j} = \langle \mathbf{b}_i, \mathbf{b}^*_j \rangle / \langle \mathbf{b}^*_j, \mathbf{b}^*_j \rangle$.

Searching $W_{k,a,j_0,B}$ with GenSample

(1) Compute $\mathbf{c}$ that satisfies Eq. (5). for $j = 1, \ldots, n$.

(2) Compute N such that $N = |W_{k,a,j_0,B}| = \prod_{j=1}^{j_0} c_j$.

(3) Call GenSample (B with $\mu_{i,j}$, $\mathbf{c}$, x) for $x = 0, \ldots, N-1$.

Significantly, all lattice vectors in the set specified by some $\mathbf{c}' = (c'_1, \ldots, c'_n) \in \mathbb{N}^n$ and $\{0, \ldots, N'-1\}$ have a one-to-one correspondence via GenSample. Here, $N' = \prod_{j=1}^{j'_0} c'_j$ for j'_0 such that j'_0 is the last index of nonzero c'_j. We referred to Ludwig [13] for the proof.

3.2 Parameters Computed from Known VSVs

We experimentally confirmed that given the parameters computed from known VSVs for ESS, the inclusion ratio of a VSV in ESS is considerably higher than that of the SA search space in our earlier work. VSVs are not known in practical situations. We studied a method of choosing the refined parameters for ESS in situations where there are no known VSVs for a given basis in Section 4.

Experimental Results

The purpose of the search is to find at least one VSV. Therefore, we need to calculate the ratio of the number of bases B such that $W_{k,a,j_0,B}$ includes at least one VSV to that of all the bases tested. We tested 100 BKZ-reduced bases on three different types of lattices, i.e., a GGH lattice, Micciancio's GGH lattice, and an NTRU lattice. We used the same set of 100 bases for all parameter sets of (k, a, j_0). We calculated the rate for $j_0 \in \{n-4, n-3, n-2, n-1, n\}$, $k \in \{1.0, 1.1, 1.2, \ldots, 2.0\}$, and $a \in \{0.8800, 0.8805, 0.8810, \cdots, 0.9985, 0.9990\}$. The two lists in **Table 1** summarize the maximum ratio for finding a VSV for a given space size of $W_{k,a,j_0,B}$ in GGH lattices in dimension 180. The maximum ratios were computed for all (k, a, j_0) tested. We also conducted the same calculation for the SA search space as listed in the lower table. By comparing the upper parts of each table with their corresponding lower parts, we can see that ES S is more effective than the SA search space in finding a VSV. In most cases, the inclusion ratio for the same space size level is much higher for ESS than for the SA search space. Here, we mean any space size 10^r by using space size level 10^z for $z = \lceil r \rceil$.

Table 2 summarizes the maximum ratio of finding a VSV for a given space size of $W_{k,a,j_0,B}$ in Miccinacio's GGH lattices in dimension 160. We have also presented the same calculation for the SA search space listed in the lower part of the table.

Table 3 summarizes the maximum ratio of finding a VSV for a given space size of $W_{k,a,j_0,B}$ in NTRU lattices in dimension 214. We also presented the same calculation for the SA search space listed in the lower part of the table.

3.3 Exhaustive Search with Single CPU

We will present the results obtained from an exhaustive search in ESS for an individual basis in the best case on the three types of lattices to confirm ESS can be exhausted by a single CPU.

Table 1 Maximum ratios at several levels of space size with their parameter sets in dimension 180. One hundred (0.99, 20)-BKZ reduced GGH public bases were used.

ESS						
Space size level	10^{10}	10^{11}	10^{12}	10^{13}	10^{14}	10^{15}
k	1.2	1.2	1.6	1.6	1.9	1.6
a	0.97	0.973	0.963	0.966	0.961	0.9695
j_0	179	180	180	180	180	180
Ratio	12%	26%	51%	78%	97%	100%
SA search space						
u	32	35	38	42	45	47
Ratio	8%	10%	19%	29%	44%	53%

Table 2 Maximum ratios at several levels of space size with their parameter sets in dimension 160. One hundred (0.99, 20)-BKZ reduced Micciancio's GGH public bases were used. These results were presented in Fukase et al. [5].

ESS						
Space size level	10^{14}	10^{15}	10^{16}	10^{17}	10^{18}	10^{19}
k	1.7	1.6	2.0	1.7	2.0	1.8
a	0.966	0.971	0.9645	0.972	0.9685	0.9735
j_0	160	160	160	160	160	160
Ratio	14%	33%	64%	86%	99%	100%
SA search space						
u	44	49	52	55	57	62
ratio	1%	7%	8%	8%	9%	9%

Table 3 Maximum ratios at several levels of space size with their parameter sets in dimension 214. One hundred (0.99, 20)-BKZ reduced NTRU bases were used.

ESS						
Space size level	10^{13}	10^{14}	10^{15}	10^{16}	10^{17}	10^{18}
k	1.6	1.9	1.8	1.9	1,8	1.9
a	0.966	0.961	0.9655	0.966	0.9695	0.9695
j_0	214	214	214	214	214	214
Ratio	6%	18%	33%	54%	81%	96%
SA search space						
u	42	45	47	52	55	57
ratio	1%	2%	3%	7%	7%	9%

3.3.1 Experimental Results

We reduced a basis in dimension 180 by (0.99, 20)-BKZ. We called the process of reduction before the search *preprocessing* within the context we used. We then calculated the optimal parameters for ESS from known VSVs in the lattice generated by the basis. Here, the optimal parameters mean those with which $W_{k,a,j_0,B}$ includes a VSV and where the space size of $W_{k,a,j_0,B}$ is the smallest. We conducted an exhaustive search with the optimal parameters to investigate the potential of the search in ESS.

First, we tested 10 bases in GGH lattices in dimension 180 for the exhaustive search. We reduced each basis with BKZ until the space size of $W_{k,a,j_0,B}$ for the basis became small enough for the actual search. Although the space size of $W_{k,a,j_0,B}$ for some bases is still large, we could conduct an actual exhaustive search for eight of the 10 bases. The optimal parameter sets and the search time are listed in **Table 4**. The "search and preprocessing (sec.)" column indicates the search time and the preprocessing time with BKZ. We used GenSample for the search in $W_{k,a,j_0,B}$.

For comparison, we have also presented the results when only BKZ was used on the eight bases that we conducted the exhaustive search on. We reduced each basis by using BKZ with increasing β. When (0.99, 24)-BKZ reduction terminated, a VSV

Table 4 Results for exhaustive search in $W_{k,a,j_0,B}$ for 10 GGH public bases in dimension 180. Optimal k, a, and j_0 were chosen for each basis. These results were presented in Fukase et al. [5].

k	a	j_0	Space size	Search and pre-processing (sec.)	BKZ only (sec.) (min. β)
1.0	0.976	178	$10^{8.730}$	30,971	615,685 (25)
1.0	0.9775	179	$10^{9.332}$	60,541	231,105 (25)
1.0	0.9765	178	$10^{9.0309}$	22,077	666,739 (25)
1.0	0.976	180	$10^{8.730}$	26,273	80,503 (24)
1.9	0.923	180	$10^{7.077}$	5,865	724,648 (25)
1.0	0.9765	179	$10^{9.031}$	31,296	638,653 (25)
1.8	0.938	180	$10^{8.457}$	14,279	348,143 (24)
1.2	0.9645	179	$10^{7.62805}$	62,995	402,806 (25)

Table 5 Results for search in $W_{k,a,j_0,B}$ for 10 Micciancio's GGH public bases in dimension 160. Optimal k, a, and j_0 were chosen for each basis. These results were presented in Fukase et al. [5].

k	a	j_0	Space size	Search and pre-processing (sec.)	BKZ only (sec.) (min. β)
2.0	0.9445	157	$10^{8.457}$	174,066	Not terminated in 10 days (26)

Table 6 Results for exhaustive search in $W_{k,a,j_0,B}$ for 10 NTRU bases in dimension 214. Optimal k, a, and j_0 are chosen for each basis. These results were presented in Fukase et al. [5].

k	a	j_0	Space size	Search and pre-processing (sec.)	BKZ only (sec.) (min. β)
1.4	0.9675	210	$10^{9.184}$	229,227	Not terminated in 10 days (25)

had been found in two cases. If a VSV was not found, we reduced each basis by using BKZ with $\beta = 25$. The total time for BKZ reduction and β with which BKZ reduction had found a VSV is also listed in Table 4. We can see from the table that the runtime for the search in ESS was much smaller than that for BKZ reduction in all cases. This indicates the potential of the search in ESS.

Second, we tested 10 bases in Micciancio's GGH lattices in dimension 160 for the exhaustive search. The optimal parameter sets and the search time are summarized in **Table 5**. We found that Micciancio's GGH public bases in dimension 160 were slightly more difficult to reduce by BKZ reduction than GGH public bases in the same dimension 160. However, we could still obtain a VSV from one of the 10 bases. BKZ, on the other hand, could not find a VSV within 10 days for any bases.

Third, we tested 10 bases in NTRU lattices in dimension 214 for the exhaustive search. The optimal parameter sets and the search time are listed in **Table 6**. We could also still find a VSV from one of 10 bases. However, BKZ could not find a VSV within 10 days for any bases.

We confirmed from the results above that it was possible to find a VSV by using an exhaustive search with a single CPU in high-dimensional lattices if the parameters were optimal.

4. Parameter Refinement

We explain how to compute the refined parameters for ESS of a given basis in this section. Here, the refined parameters mean those that maximize the inclusion probability of a VSV under a given space size. Thus, we must compute the inclusion probability of a VSV to refine the parameters. We consider the deviation of a VSV from its expected value on the Gram-Schmidt coefficients as a probabilistic variable to compute the inclusion probability. However, there are no known VSVs for a given basis, and it is difficult to estimate the probabilistic distributions of the variable on which the computation of the inclusion probability is based from the basis alone. Therefore, we utilize other bases where VSVs are known to obtain some probabilistic distributions. This is possible because the key generation algorithms are known in some cryptographic situations. Assuming such cryptographic situations, we introduce a scheme PR that computes the inclusion probability by utilizing many bases besides its input basis and it outputs the parameters for ESS maximizing the inclusion probability in Section 4.2. Some particular distributions are exploited in GGH lattices and NTRU lattices because these types of lattices have some specific structures. We targeted GGH lattices and NTRU lattices to demonstrate the performance of PR, which is discussed in Sections 4.3 and 4.4. Compared with GGH lattices and NTRU lattices, Micciancio's GGH lattices do not have any specific structure, and in fact the distributions are dependent on bases. Consequently, the inclusion probability computed with the average distribution is not reliable. For such situations, we consider another approach to refining parameters instead of computing the inclusion probability. Section 4.5 explains how weaker refinement of parameters is possible by using some empirical approaches.

4.1 Inclusion Probability Analysis

We utilize the deviation of a VSV from its expected value on the Gram-Schmidt coefficients and that of a basis from GSA to compute the inclusion probability of a VSV. The following analysis is an extended version of that in Fukase and Yamaguchi [4].

We first consider the expected value of the Gram-Schmidt coefficients of VSVs to describe the deviation of a VSV. Let $\mathbf{v} = \sum_{j=1}^{n} v_j \mathbf{b}_j^*$ be a VSV in the lattice generated by basis B, and let $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$ be the Gram-Schmidt orthogonalized vectors of B. Here, we assume that B has been reduced by LLL or BKZ. Because $\mathbf{v}$ is a VSV, each v_j is expected to be small so that v_j^2 cancels the term, $\|\mathbf{b}_j^*\|^2$. We introduce the following heuristic assumption

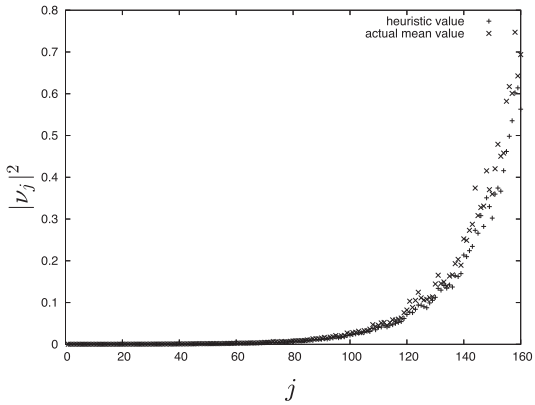

Fig. 3 Heuristic values and actual mean values of $|\nu_j|^2$. Values of $|\nu_j|^2$ are averaged on 160 VSVs for actual mean values.

based on this idea.

Assumption 1 Let ν_j for $j = 1, \ldots, n$ represent the Gram-Schmidt coefficients of any VSVs in a lattice, $L(B)$. Then,

$$E[\,|\nu_j|^2\,] = t^2/\|\mathbf{b}_j^*\|^2 \quad \text{for} \quad j = 1, \ldots, n \tag{6}$$

holds for some constant $t \in \mathbb{R}_+$.

Although Assumption 1 does not hold rigorously, it is sufficiently close so that it makes sense to assume that it is true. For example, see **Fig. 3**.

Figure 3 plots $E[\,|\nu_j|^2\,]$ calculated based on Assumption 1 and the actual mean values of $|\nu_j|^2$ for a basis in dimension 160. The values of $|\nu_j|$ are averaged on 160 VSVs for the actual mean values. Here, t is determined as follows. Because a VSV is expected to be as short as λ_1, Assumption 1 states that $\lambda_1^2 \approx E[\,\|\mathbf{v}\|^2\,] = E[\,\sum_{j=1}^n \nu_j^2\|\mathbf{b}_j^*\|^2\,] = \sum_{j=1}^n (t^2/\|\mathbf{b}_j^*\|^2)\|\mathbf{b}_j^*\|^2 = nt^2$. Therefore, it is reasonable to use $\lambda_1/n^{1/2}$ as t in Eq. (6).

In the following, we represent $|\nu_j|$ with t and $|\mathbf{b}_j^*|$ for all $j = 1, \ldots, n$ as

$$|\nu_j| = (t + t_j)/\|\mathbf{b}_j^*\| \tag{7}$$

for a probabilistic variable, $t_j \in \mathbb{R}$. Now, we consider $|\nu_j| = t/\|\mathbf{b}_j^*\|$ as the expected value of $|\nu_j|$ and t_j as the deviation of $|\nu_j|$ from the value.

Next, we consider the deviation of a basis from GSA. For that, we represent $\|\mathbf{b}_j^*\|$ with another form. Recall that if basis B is reduced by LLL or BKZ, the lengths of the Gram-Schmidt orthogonalized vectors of the basis resemble a geometric sequence, $\|\mathbf{b}_j^*\|^2 \approx q^{j-1}\|\mathbf{b}_1\|^2$, for some common ratio $q \in [0, 1]$. In our experiment, we compute q with the method of least mean squares so that $\sum_{j=1}^n (\|\mathbf{b}_j^*\|^2 - q^{j-1}\|\mathbf{b}_1\|^2)^2$ is minimum. We explicitly describe the deviation of the approximate equation, $\|\mathbf{b}_j^*\|^2 \approx q^{j-1}\|\mathbf{b}_1\|^2$, as in the equation

$$\|\mathbf{b}_j^*\|^2 = e^{\delta_j} q^{j-1}\|\mathbf{b}_1\|^2 \quad \text{for} \quad j = 1, \ldots, n, \tag{8}$$

with error terms δ_j for $j = 1, \ldots, n$ whose absolute values are supposed to be small.

Now, we represent $|\nu_j|$ with the deviations, t_j and δ_j. From Eqs. (7) and (8):

$$\begin{aligned}
|\nu_j| &= (t + t_j)/(e^{\delta_j} q^{j-1}\|\mathbf{b}_1\|^2)^{1/2} \\
&= t/(q^{j-1}\|\mathbf{b}_1\|^2)^{1/2} \\
&\quad +((1 - e^{\delta_j/2})t + t_j)/(e^{\delta_j} q^{j-1}\|\mathbf{b}_1\|^2)^{1/2} \\
&= (t(q^{1/2})^{1-n}/\|\mathbf{b}_1\|)(q^{1/2})^{n-j} \\
&\quad +((1 - e^{\delta_j/2})t + t_j)/(e^{\delta_j} q^{j-1}\|\mathbf{b}_1\|^2)^{1/2}.
\end{aligned}$$

Let $\bar{k} = t(q^{1/2})^{1-n}/\|\mathbf{b}_1\|$, $\bar{a} = q^{1/2}$, and $\epsilon_j = ((1 - e^{\delta_j/2})t + t_j)/(e^{\delta_j} q^{j-1}\|\mathbf{b}_1\|^2)^{1/2}$. Then,

$$|\nu_j| = \bar{k}\bar{a}^{n-j} + \epsilon_j. \tag{9}$$

Here, let $k = \bar{k}$, $a = \bar{a}$, and j_0 be parameters for ESS $W_{k,a,j_0,B}$. From Definition 1, Eq. (9), and $\epsilon_j = ((1 - e^{\delta_j/2})t + t_j)/(e^{\delta_j} q^{j-1}\|\mathbf{b}_1\|^2)^{1/2}$, the condition below needs to be satisfied for $\mathbf{v} = \sum_{j=1}^n \nu_j \mathbf{b}_j^*$ to be included in ESS:

Condition 1

$$t_j \le (\lceil 2ka^{n-j} \rceil/2 - ka^{n-j})(e^{\delta_j} q^{j-1}\|\mathbf{b}_1\|^2)^{1/2} - (1 - e^{\delta_j/2})t \quad \text{for } 1 \le j < j_0, \tag{10a}$$

$$t_j \le (\lceil ka^{n-j} \rfloor - ka^{n-j})(e^{\delta_j} q^{j-1}\|\mathbf{b}_1\|^2)^{1/2} - (1 - e^{\delta_j/2})t \quad \text{for } j = j_0, \tag{10b}$$

$$t_j = -ka^{n-j}(e^{\delta_j} q^{j-1}\|\mathbf{b}_1\|^2)^{1/2} - (1 - e^{\delta_j/2})t \quad \text{for } j_0 < j \le n. \tag{10c}$$

Let $\bar{t}_j = (\lceil 2ka^{n-j} \rceil/2 - ka^{n-j})(e^{\delta_j} q^{j-1}\|\mathbf{b}_1\|^2)^{1/2} - (1 - e^{\delta_j/2})t$ and $\bar{t}_{j_0} = (\lceil ka^{n-j} \rfloor - ka^{n-j})(e^{\delta_j} q^{j-1}\|\mathbf{b}_1\|^2)^{1/2} - (1 - e^{\delta_j/2})t$. Also, let $p_{j_0,j}$ be the cumulative distribution function of t_j. Then, the inclusion probability, p, is:

$$p = \left(\prod_{j=1}^{j_0-1} p_{j_0,j}(\bar{t}_j)\right) p_{j_0,j_0}(\bar{t}_{j_0}). \tag{11}$$

4.2 Sample Bases Approach to Parameter Refinement

The following subsection presents a scheme to compute the refined parameters for ESS, whereby the inclusion probabilities for various sets of parameters for ESS are computed using the formula to compute the inclusion probability in Section 4.1. As stated in Section 4.1, we need the probabilistic distribution of the probabilistic variable, t_j, to compute the inclusion probability. Therefore, we use the following information available from bases where VSVs are known. We call such bases *training bases*.

The cumulative distribution function, $p_{j_0,j}$, of t_j is computed on all $\mathbf{v} = \sum_{j=1}^n \nu_j \mathbf{b}_j^*$ such that j_0 is the last index of nonzero coefficient ν_j. Note that $p_{j_0,j}(x) = 0$ for $x < -t$ because $|\nu_j| = (t + t_j)/\|\mathbf{b}_j^*\| \ge 0$, and $p_{j_0,j}(\infty) = 1$.

Let p_{j_0} be the fraction of all $\mathbf{v}$ in training bases such that j_0 is the last index of nonzero coefficient ν_j. Also, let λ_1' be the mean value of $\|\mathbf{v}\|$ on training bases, and let $\underline{j_0}$ be the possible minimum value of the last index, j_0, of nonzero coefficient ν_j.

These inputs are used to calculate the refined values of k, a, and j_0 in the following algorithm PR.

Parameter Refinement for ESS (PR)

Input:

B lattice basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$

λ_1' the mean value of $\|\mathbf{v}\|$ on training bases

$\underline{j_0}$ the minimum value of j_0 on training bases

$p_{j_0,j}$ cumulative distribution function of t_j on all $\mathbf{v}$ specified by j_0
p_{j_0} fraction of all $\mathbf{v}$ on training bases where $\mathbf{v}$ is specified by j_0
α, d parameters to determine q'
s space size bound

Output:

P the probability with which at least one VSV exists in ESS.
(k, a, j_0) the refined parameters for ESS.

Procedure:

compute $\|\mathbf{b}_1^*\|^2, \ldots, \|\mathbf{b}_n^*\|^2$
compute q s.t. $\|\mathbf{b}_j^*\|^2 \approx q^{j-1}\|\mathbf{b}_n\|^2$ with the method of least mean squares
compute $t := \lambda_1'/n^{1/2}$
$P := 0$
for $z = -d, \ldots, 0, \ldots, d$
  $q' := q + \alpha z$
  $a' := q'^{1/2}$
  $k' := t(q'^{1/2})^{1-n}/\|\mathbf{b}_1\|$
  for $j = 1, \ldots, n$
    $\delta_j' := 2\log(\|\mathbf{b}_j^*\|) - 2\log(\|\mathbf{b}_1\|) - (j-1)\log(q')$
  for $j_0' = \underline{j_0}, \ldots, n$
    $p := p_{j_0'}$
    for $j = 1, \ldots, j_0'$
      if $(j < j_0')$
        $p := p\, p_{j_0',j}(\bar{t}_j)$
      else
        $p := p\, p_{j_0',j}(\bar{t}_{j_0})$
    $P' := 1 - (1-p)^n$
    if $(P' > P$ and $|W_{k',a',j_0',B}| \leq s)$
      $(k, a, j_0) := (k', a', j_0')$
      $P := P'$

4.3 Sample Bases Approach to Parameter Refinement on GGH Lattices

We present the performance of PR on GGH lattices here. GGH lattices have specific structures, and therefore some particular distributions of t_j are obtained.

First, we generated 100 $(0.99, 20)$-BKZ reduced GGH bases as training bases and calculated λ_1', $\underline{j_0}$, p_{j_0,t_j}, and p_j from them. In this experiment, $\lambda_1' = 69.22$ and $\underline{j_0} = 167$.

We computed the refined parameters for the 100 $(0.99, 20)$-BKZ reduced GGH bases in dimension 180 used in Table 1 with PR. One hundred bases were generated independently of training bases. In PR, q' is searched around q, and the candidate set of $q's$ is $\{q-\alpha d, q-\alpha(d-1), \ldots, q, \ldots, q+\alpha(d-1), q+\alpha d\}$ with α and d. Note that q differs according to the bases. For example, we obtained $q = 0.948$ for a basis. With $\alpha = 0.0001$ and $d = 80$, the candidate set of $q's$ was $\{0.940, 0.9401, \ldots, 0.9559, 0.956\}$. For s, we used 10^{11}, 10^{12}, 10^{13}, 10^{14}, and 10^{15}. **Table 7** summarizes the results obtained from the experiment. The "ratio" column in Table 7 indicates the ratios with which at least one VSV was actually included in ESS for the refined parameters.

We used the same 100 bases to calculate Tables 1 and 7. The ratios in Table 1 were calculated by knowing the VSVs. Those in Table 7, on the other hand, were calculated using the refined parameters. In Table 7, $\frac{\text{ratio (PR)}}{\text{ratio (known VSVs)}}$ is the goodness of the refined parameters (PR) compared to the parameters computed from known VSVs in Table 1. "Ratio (PR)" is the ratio of the refined parameters. "Ratio (known VSVs)" represents the parameters computed from known VSVs. For example, the former is 11% in Table 7, and the latter is 12% in Table 1. Therefore, $\frac{\text{ratio (PR)}}{\text{ratio (known VSVs)}} = 0.92$. In Table 7, $\frac{\text{ratio (PR)}}{\text{ratio (known VSVs)}}$ is high between 10^{10} and 10^{15} and especially at 10^{10}, 10^{11}, and 10^{15}, it is close to 1.0. Thus, it can be said that the results in Table 7 are close to those in Table 1. This means that PR performs well.

Table 7 Inclusion ratios for refined parameters. One hundred $(0.99, 20)$-BKZ reduced GGH public bases in dimension 180 were used.

s	10^{10}	10^{11}	10^{12}	10^{13}	10^{14}	10^{15}
Ratio	11%	25%	37%	60%	73%	100%
$\frac{\text{Ratio (PR)}}{\text{Ratio (known VSVs)}}$	0.92	0.96	0.73	0.78	0.75	1.00

We also tested some sets of training bases to investigate how many bases were sufficient for training bases. We confirmed that less than 100 bases or bases with weaker reductions were permissible for training bases. However, the results for 10 bases were unstable. Therefore, we considered 10 bases for training bases to be insufficient.

4.4 Sample Bases Approach to Parameter Refinement on NTRU Lattices

We introduce the performance of PR on NTRU lattices here. NTRU lattices also have specific structures, and therefore some particular distributions of t_j are obtained.

However, we need slight modifications to apply PR to NTRU lattices because $\|\mathbf{b}_j^*\|^2$ for $j = 1, \ldots, n$ do not approximate GSA for NTRU bases reduced by BKZ with moderate β. The initial lengths, $\|\mathbf{b}_1^*\|^2, \ldots, \|\mathbf{b}_{h-1}^*\|^2$, for some h deviate badly from a geometric sequence. However, $\|\mathbf{b}_j^*\|^2$ for $j \geq h$ approximates a geometric sequence. Moreover, $|\nu_j|$ for $j < h$ could not be larger than 0.5 because the lengths, $\|\mathbf{b}_1^*\|^2, \ldots, \|\mathbf{b}_{h-1}^*\|^2$, were relatively long. Therefore, in such cases, we can safely ignore initial $\|\mathbf{b}_1^*\|^2, \ldots, \|\mathbf{b}_{h-1}^*\|^2$ by replacing the vector index of $\mathbf{b}_1^*$ with that of $\mathbf{b}_h^*$. The t must be determined to be $t = (\lambda_1^2 - \sum_{j=1}^{h-1} \nu_j^2\|\mathbf{b}_j^*\|^2)^{1/2}/(n-(h-1))^{1/2}$ based on the argument in Section 4.1. The value of the term, $\sum_{j=1}^{h-1} \nu_j^2\|\mathbf{b}_j^*\|^2$, in the above form cannot easily be estimated. However, we experimentally confirmed that the contribution of the term, $\sum_{j=1}^{h-1} \nu_j^2\|\mathbf{b}_j^*\|^2$, and $(h-1)$ to the value of t was small. Consequently, we replaced $(\lambda_1^2 - \sum_{j=1}^{h-1} \nu_j^2\|\mathbf{b}_j^*\|^2)^{1/2}/(n-(h-1))^{1/2}$ with $\lambda_1/n^{1/2}$ by just ignoring them. As a result, we have set t to $\lambda_1/n^{1/2}$ here. We tested the bases where VSVs were known to verify this approximation. In the experiment in dimension 214, we obtained $\lambda = 59.0$, $\sum_{j=1}^{h-1} \nu_j^2\|\mathbf{b}_j^*\|^2 = 5.0$ and $h - 1 = 17$, then $(\lambda_1^2 - \sum_{j=1}^{h-1} \nu_j^2\|\mathbf{b}_j^*\|^2)^{1/2}/(n-(h-1))^{1/2} = 0.526$ and $\lambda_1/n^{1/2} = 0.525$ for example. Thus, $\lambda_1/n^{1/2}$ sufficiently approximates $(\lambda_1^2 - \sum_{j=1}^{h-1} \nu_j^2\|\mathbf{b}_j^*\|^2)^{1/2}/(n-(h-1))^{1/2}$.

Let q be the common ratio of a geometric sequence that $\|\mathbf{b}_j^*\|^2$ for $j \geq h$ approximates, and δ_j such that $\|\mathbf{b}_j^*\|^2 = e^{\delta_j} q^{j-h}\|\mathbf{b}_h\|^2$ for $j = h, \ldots, n$, $k = t(q^{1/2})^{h-n}/\|\mathbf{b}_h\|$, and $a = q^{1/2}$. Let $\mathbf{v} = \sum_{j=1}^{n} \nu_j \mathbf{b}_j^*$ be a VSV in an NTRU lattice, and t_j such that

$|\nu_j| = (t + t_j)/\|\mathbf{b}_j^*\|$. Then, in this case, Condition 1 is rewritten into Condition 2 below:

Condition 2

$$t_j \le (\lceil 2ka^{n-j} \rceil/2 - ka^{n-j})(e^{\delta_j} q^{j-h} \|\mathbf{b}_h\|^2)^{1/2} - (1 - e^{\delta_j/2})t \quad \text{for } h \le j < j_0, \tag{12a}$$

$$t_j \le (\lceil ka^{n-j} \rfloor - ka^{n-j})(e^{\delta_j} q^{j-h} \|\mathbf{b}_h\|^2)^{1/2} - (1 - e^{\delta_j/2})t \quad \text{for } j = j_0, \tag{12b}$$

$$t_j = -ka^{n-j}(e^{\delta_j} q^{j-h} \|\mathbf{b}_h\|^2)^{1/2} - (1 - e^{\delta_j/2})t \quad \text{for } j_0 < j \le n. \tag{12c}$$

We applied modified PR to NTRU lattices in dimension 214. As was discussed in Section 4.3, we experimentally investigated how many bases were sufficient for training bases. We observed that 100 bases were sufficient as training bases and very few bases, e.g., 20 bases, occasionally were not sufficient where PR was applied to NTRU lattices. Subsequently, we used 100 bases as training bases.

Here, we explain our evaluation of the performance of PR on NTRU lattices. First, we generated 100 $(0.99, 20)$-BKZ reduced NTRU bases as training bases and calculated λ_1', $\underline{j_0}$, p_{j_0,t_j}, and p_j from them. In this experiment, $\lambda_1' = 7.68$ and $\underline{j_0} = 203$. We computed the refined parameters for 100 $(0.99, 20)$-BKZ reduced NTRU bases in the dimension 214 used in Table 3 with PR. One hundred bases were generated independently of the training bases. We obtained $q = 0.949$ for some bases. With $\alpha = 0.0001$ and $d = 120$, the candidate set of $q's$ was $\{0.937, 0.9371, \ldots, 0.9609, 0.961\}$. For s, we used 10^{14}, 10^{15}, 10^{16}, 10^{17}, and 10^{18}. **Table 8** summarizes the results obtained from this experiment. The "ratio" column in Table 8 indicates ratios with which at least one VSV was actually included in ESS for the refined parameters.

We used the same 100 bases as in Tables 3 and 8. In Table 8, $\frac{\text{ratio (PR)}}{\text{ratio (known VSVs)}}$ is the goodness of the refined parameters (PR) compared with the parameters computed from known VSVs in Table 3. "Ratio (PR)" is the ratio of the refined parameters. "Ratio (known VSVs)" represents the parameters computed from known VSVs. For example, the former is 3% in Table 8, and the latter is 6% in Table 3. Therefore, $\frac{\text{ratio (PR)}}{\text{ratio (known VSVs)}} = 0.50$. From 8, $\frac{\text{ratio (PR)}}{\text{ratio (known VSVs)}}$ is more than 1.00 in some cases. This means that at some space size levels, PR achieved higher ratios than those for the experiment listed in Table 3. This was possible because parameters were individually determined for each input basis with PR while the parameters computed from known VSVs were applied equally to all targeted bases. Thus, PR performed well.

4.5 Empirical Approach to Parameter Refinement

Micciancio's GGH lattices do not have any specific structure, and the distributions are dependent on bases. Because the average distribution on many bases concerning Micciancio's GGH lattices is useless for computing the inclusion probability, PR cannot be applied to Micciancio's GGH lattices. Even in such cases, it is possible to refine parameters with some empirical approach because there are not that many candidates for a proper parameter set in practice. As we saw in Section 4.4, the parameters refined by PR are sometimes better than the parameters computed from known VSVs. However, the parameters computed with parameter refinement presented here can never be better than the parameters computed from known VSVs. In that sense, the parameter refinement presented here is weaker.

First, consider the choice of j_0. See **Fig. 4**. Figure 4 plots the inclusion ratios for the 100 bases at several levels of space size for $j_0 = 158$, $j_0 = 159$, and $j_0 = 160$. The k in the figure is fixed at 2.0, and it can be seen that the inclusion ratios for $j_0 = n$ are much higher than those for j_0 smaller than n.

Second, consider the choice of k. **Figure 5** plots the inclusion ratios for the 100 Micciancio's GGH bases used in Table 2 at several levels of space size for $k = 1.0$, $k = 1.5$, and $k = 2.0$. The j_0 in the figure is fixed at 160. The inclusion ratios in Table 2, which were achieved when VSVs were known, have been shown for comparison. As seen in Fig. 5, $k = 1.0$ achieves much lower inclusion ratios than other ks, while any $k > 1.0$ achieves similarly high inclusion ratios. Hence, $k = 1.0$ should be avoided.

From the above observation, one can set j_0 to n and k to 2.0.

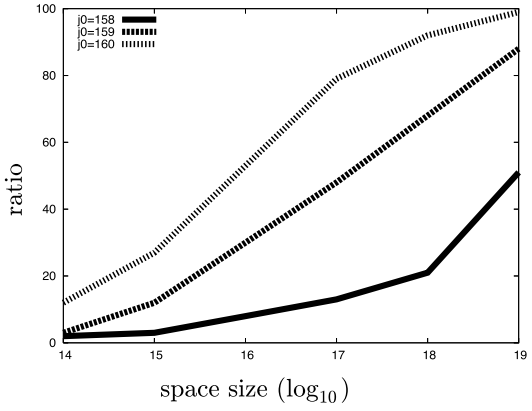

Fig. 4 Inclusion ratios for 100 Micciancio's GGH bases for $j_0 = 158$, $j_0 = 159$, and $j_0 = 160$. k was fixed at 2.0.

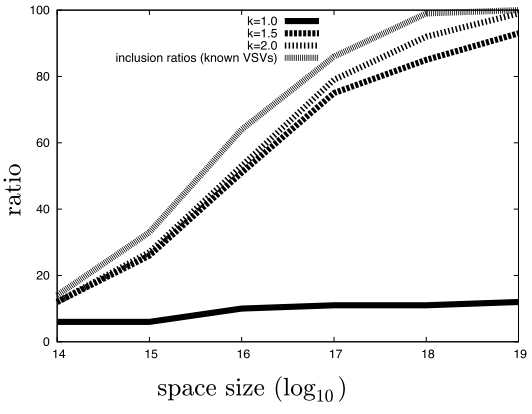

Fig. 5 Inclusion ratios for 100 Micciancio's GGH bases for $k = 1.0$, $k = 1.5$, and $k = 2.0$. j_0 was fixed at 160. Inclusion ratios in Table 2 have been shown.

Table 8 Inclusion ratios for refined parameters. One hundred $(0.99, 20)$-BKZ reduced NTRU bases in dimension 214 were used.

s	10^{13}	10^{14}	10^{15}	10^{16}	10^{17}	10^{18}
Ratio	3%	7%	31%	63%	86%	97%
$\frac{\text{Ratio (PR)}}{\text{Ratio (known VSVs)}}$	0.50	0.39	0.94	1.17	1.06	1.01

Then, a is determined by a given space size with $j_0 = n$ and $k = 2.0$. In summary, the parameter refinement here, which we call empirical refinement, is as follows:

Empirical Refinement

(1) set j_0 to n

(2) set k to 2.0

(3) compute a from given space size with j_0 and k

We concluded that a proper parameter set, which is only a little worse than the parameter set computed from known VSVs, could be determined by using empirical refinement. We confirmed empirical refinement could also be applied to GGH lattices and NTRU lattices when training bases were not available.

5. Exhaustive Search with Multiple CPUs

This section introduces a method of distributed search and presents the results obtained from the distributed search in ESS on eight CPUs.

Because sampling reduction does not manipulate the basis during the sampling phase at all, it is subject to easy distribution. Ludwig proposed a method of parallelizing sampling reduction by running GenSample again and again on each CPU, and he estimated how effective the method was [13]. Here, we confirm how effective the method in ESS is with an actual search.

We need to employ many CPUs to conduct a distributed search in ESS. Suppose there are z CPUs available. As explained in Section 3.1, all lattice vectors in $W_{k,a,j_0,B}$ and $\{0,\ldots,N-1\}$ with $N = |W_{k,a,j_0,B}|$ have a one-to-one correspondence by using GenSample. We partition the search space in disjunct parts $W_{k,a,j_0,B} = \cup_{j=1}^{z} W_j$ where all lattice vectors in W_j and $\{(\lfloor N/z \rfloor)(j-1),\ldots,(\lfloor N/z \rfloor)j-1\}$ have a one-to-one correspondence via GenSample for $1 \le j \le z-1$ and all lattice vectors in W_j and $\{(\lfloor N/z \rfloor)(j-1),\ldots,N-1\}$ have a one-to-one correspondence via GenSample for $j = z$. Then, we run GenSample for W_j on the j-th CPU for $j = 1,\ldots,z$. We can expect that the distributed search in ESS will reduce the search time by at least z.

We conducted small scale experiments on a distributed search in ESS. We used eight CPUs. We conducted the distributed search in ESS on three types of lattices as described in Section 3. We used the same bases and corresponding optimal parameters as in Section 3. The results are summarized in **Tables 9**, **10**, and **11**, where the search time does not include the preprocessing time with BKZ.

In most cases, the distributed search in ESS reduced the search time by a factor that was much larger than z. This phenomenon is strange because we expected that the search time would be reduced by factor z on average. A possible explanation for this phenomenon is as follows. We partitioned the search space into disjunct parts $W_{k,a,j_0,B} = \cup_{j=1}^{8} W_j$ in the experiments. We confirmed that in many cases a VSV existed in W_j with a relatively large j. This seemed to be caused by the parameters selected for ESS. We chose parameters with which $W_{k,a,j_0,B}$ included a VSV and the space size of $W_{k,a,j_0,B}$ was the smallest. A VSV for such parameters possibly fell in near the end of $W_{k,a,j_0,B}$. Furthermore, we assumed that a VSV would accidentally fall in near the start of some W_j in the experiments.

Table 9 Results from distributed search in $W_{k,a,j_0,B}$ for eight GGH public bases in dimension 180. Eight CPUs were used.

Search time sec. (1 CPU)	Distributed search time sec. (8 CPUs)
6,890	561
55,872	4,305
10,552	13
19,853	12
337	52
22,178	843
9,867	222
4,142	105

Table 10 Results from distributed search in $W_{k,a,j_0,B}$ for Micciancio's GGH public basis in dimension 160. Eight CPUs were used.

Search time sec. (1 CPU)	Distributed search time sec. (8 CPUs)
7,544	91

Table 11 Results from distributed search in $W_{k,a,j_0,B}$ for NTRU basis in dimension 214. Eight CPUs were used.

Search time sec. (1 CPU)	Distributed search time sec. (8 CPUs)
165,875	4,202

6. Conclusion

We enabled parameter refinement for ESS and confirmed the distributed search in ESS was effective by achieving significant speedups. The phenomenon we observed in Section 5 may be exploited to improve our method of search. We found that in some cases the search time was reduced by a factor that was larger than 1,000 on only eight CPUs. Such significant speedups might be caused on purpose by inventing some proper order for the search rather than just forward or backward searches. This direction of research may offer further suggestions to advance studies on exhaustive search.

References

[1] Ajtai, M. and Dwork, C.: A Public-Key Cryptosystem with worst-case/average-case equivalence, *Proc. 29th STOC*, pp.284–293 (1997).

[2] Ajtai, M.: The Shortest Vector Problem in L_2 is NP-hard for Randomized Reductions (Extended Abstract), *Proc. Thirtieth Annual ACM Symposium on Theory of Computing*, pp.10–19 (1998).

[3] Dagdelen, Ö. and Schneider, M.: Parallel Enumeration of Shortest Lattice Vectors, *Euro-Par 2010*, Vol.6272 of LNCS, pp.211–222 (2010).

[4] Fukase, M. and Yamaguchi, K.: The Analysis of ESS for the Shortest Vector in Lattice, *Proc. SICT 2010*, pp.209–213 (2010).

[5] Fukase, M. and Yamaguchi, K.: Exhaustive Search for Finding a Very Short Vector in High-Dimensional Lattices, *Proc.* (*short papers*) *IWSEC 2010*, pp.26–41 (2010).

[6] Gama, N. and Nguyen, P.Q.: Predicting Lattice Reduction, *EUROCRYPT 2008*, Vol.4965 of LNCS, pp.31–51 (2008).

[7] Gama, N., Nguyen, P.Q. and Regev, O.: Lattice Enumeration Using Extreme Pruning, *EUROCRYPT 2010*, Vol.6110 of LNCS, pp.257–278 (2010).

[8] Goldreich, O., Goldwasser, S. and Halevi, S.: Public-Key Cryptosystems from Lattice Reduction Problems, *Advances in Cryptology - Crypto'97*, Vol.1294 of LNCS, pp.112–131, Springer-Verlag (1997).

[9] Hermans, J., Schneider, M., Buchmann, J., Vercauteren, F. and Preneel, B.: Parallel Shortest Lattice Vector Enumeration on Graphics Cards, *AFRICACRYPT 2010*, Vol.6055 of LNCS, pp.52–68 (2010).

[10] Hoffstein, J., Pipher, J. and Silverman, J.H.: NTRU: A Ring-Based Public Key Cryptosystem, *Proc. ANTS III*, Vol.1423 of LNCS, pp.267–288, Springer-Verlag (1998).

[11] Kuo, P., Schneider, M., Dagdelen, Ö., Buchmann, J., Cheng, C. and Yang, B.: Extreme Enumeration on GPU and in Clouds - How Many Dollars You Need to Break SVP Challenges, *CHES 2011*, Vol.6917 of LNCS, pp.160–175 (2011).

[12] Lenstra, A.K., Lenstra, H.W. and Lovász, L.: Factoring Polynomials

with Rational Coefficients, *Mathematische Ann.*, Vol.261, pp.513–534 (1982).
[13] Ludwig, C.: Practical Lattice Basis Sampling Reduction, PhD thesis, TU Darmstadt (2005), available from ⟨http://elib.tu-darmstadt.de/diss/000640/⟩ (accessed 2011-02-28).
[14] Micciancio, D.: Improving Lattice Based Cryptosystems Using the Hermite Normal Form, *Silverman*, pp.126–145 (2001).
[15] Micciancio, D.: The Shortest Vector Problem is NP-hard to approximate to Within Some Constant, *SIAM Journal on Computing*, Vol.30, No.6, pp.2008–2035 (2001).
[16] Nguyen, P.Q. and Stehlé, D.: LLL on the Average, *ANTS*, pp.238–256 (2006).
[17] Schnorr, C.P. and Euchner, M.: Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems, *Math. Programming*, Vol.66, pp.181–199 (1994).
[18] Schnorr, C.P.: Lattice Reduction by Random Sampling and Birthday Methods, *STACS 2003*, Vol.2607 of LNCS, pp.145–156, Springer-Verlag (2003).
[19] Shoup, V.: NTL - A Library for Doing Number Theory, available from ⟨http://www.shoup.net/ntl/index.html⟩.

Masaharu Fukase received his B.S., M.S., and Ph.D. degrees from the University of Tokyo, Japan, in 2006, 2008, and 2011, respectively. Since 2011, he has been an project research assistant of Dokkyo University, Japan. His research interests include Lattices and Cryptography.

Kazunori Yamaguchi received his B.S., M.S., and Doctor of Science degrees in information science from the University of Tokyo, in 1979, 1981, and 1985, respectively. Currently, he is a professor of the University of Tokyo. His research interest is in data models and data analysis.