Random Integer Lattice Generation via HNF *

Gengran Hu[1], Lin You[2], Liqin Hu[2], and Hui Wang[2]

[1] School of Cyberspace, Hangzhou Dianzi University, Hangzhou, 310018, China;
State Key Laboratory of Information Security, Institute of Information Engineering,
Chinese Academy of Sciences, Beijing, 100093, China
grhu@hdu.edu.cn
[2] School of Cyberspace, Hangzhou Dianzi University, Hangzhou, 310018, China
mryoulin@gmail.com, huliqin@hdu.edu.cn, h.wang@hdu.edu.cn

Abstract. Lattices used in cryptography are integer lattices. Defining and generating a "random integer lattice" are interesting topics. A generation algorithm for random integer lattice can be used to serve as a random input of all the lattice algorithms. In this paper, we recall the definition of random integer lattice given by G.Hu et al. and present an improved generation algorithm for it via Hermite Normal Form. It can be proved that with probability ≥ 0.99, this algorithm outputs an n-dim random integer lattice within $O(n^2)$ operations.

Keywords: random integer lattice, Hermite Normal Form, generation algorithm

1 Introduction

Lattices are discrete subgroups in $\mathbb{R}^n$. Since M.Ajtai's discovery of the average-case/worst-case connection in lattice problems [1], lattice-based cryptography has attracted much attention [2, 10, 11, 20]. Up to now, lattice-based cryptographic schemes have been considered to be a promising alternative to more traditional ones based on the factoring and discrete logarithm problems since lattice-based schemes have yet not to be broken by efficient quantum algorithms[22]. Lattice algorithms like LLL[13] and BKZ[4, 14] are commonly used in analyzing these lattice-based schemes' security. The lattices used in cryptography and lattice algorithms are **integer lattices**(discrete subgroups of $\mathbb{Z}^n$). Thus the problem of suitably defining and generating a random integer lattice is a meaningful topic. In [18], P.Q. Nguyen found that for dimension up to 50, LLL almost outputs the shortest lattice vector, while in theory LLL's output is just an approximately short vector. Once we are able to generate random integer lattice, such an generation algorithm can be used to serve as a random input for all lattice algorithms to obtain their output qualities on the average.

In [1], M.Ajtai defined a family of "random integer lattice" in terms of worst-case to average-case connection and shows how to generate one from this lattice

* This work was supported in part by the National Natural Science Foundation of China (No.61602143, No.61602144, No.61772166) and in part by the Key Program of the Nature Science Foundation of Zhejiang province of China (No.LZ17F020002).

family. For uniform $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, the lattice family is defined to be $\Lambda^{\perp}(\mathbf{A}) = \{\mathbf{Ax} \in \mathbb{Z}^m : \mathbf{Ax} = \mathbf{0} \in \mathbb{Z}_q^n\}$. In [18], P.Q. Nguyen gave a definition of "random integer lattice" via Hermite Norm Form(HNF). For large prime number P, this random integer lattice is defined to be uniformly chosen from the set of $n \times n$ Hermite Normal Forms with $h_{nn} = P$, $h_{in} \in [0, P)$ and $h_{ii} = 1$ for $i < n$. This definition directly gives the lattice basis but the lattice determinant P needs to be a prime.

In [17], G.Maze studied the probabilistic distribution of random HNF with special diagonal structure, where the randomness is from "natural density". In [21], G.Hu et al. introduced a different definition of randomness, in which the definition "random integer lattice" means the lattice's HNF is chosen uniformly from the all $n \times n$ HNFs with determinant upper bounded by a large number M. In the same paper [21], G.Hu et al. also presented a complete random integer lattice generation algorithm. In this algorithm, the first step is to generate a determinant. To make the final output be uniform, it is necessary to compute the total number of HNFs with fixed determinant N. Since the total number can be figured out only in the case that the factorization of N is known, a subroutine to factor integers is necessary in this algorithm. While in this paper, we improved this algorithm with the help of diagonal elements' distribution in random HNF. This improved algorithm first generates the diagonal elements $h_{11}, \cdots, h_{n-1,n-1}$ without computing the total number of HNFs with fixed determinant, then it uses the reverse sampling method to generate the final diagonal element h_{nn}. Thus the factorization subroutine is no longer needed in this improved one, making it a more efficient algorithm.

Roadmap. The remainder of the paper is organized as follows. In Section 2, we give some preliminaries needed. In Section 3, we recall the definition of random integer lattice given by G.Hu et al. and discuss the distribution of all the elements in random integer lattice's HNF. For the next section, we present our improved algorithm to generate random integer lattice via HNF. Finally, we give our conclusion in Section 5.

2 Preliminaries

We denote by $\mathbb{Z}$ the integer ring and $\mathbb{R}$ the real number field. We use $GL_n(\mathbb{Z})$ to denote the general linear group over $\mathbb{Z}$. For convenience, we denote the set of all the $n \times n$ nonsingular integer matrices by $GL_n(\mathbb{R}) \cap \mathbb{Z}^{n\times n}$.

2.1 Lattice and HNF

Given a matrix $B = (b_{ij}) \in \mathbb{R}^{n\times m}$ with rank n, the lattice $\mathcal{L}(B)$ spanned by the rows of B is

$$\mathcal{L}(B) = \{xB = \sum_{i=1}^{n} x_i b_i | x_i \in \mathbb{Z}\},$$

where b_i is the i-th row of B. We call m the dimension of $\mathcal{L}(B)$ and n its rank. The determinant of $\mathcal{L}(B)$, say $\det(\mathcal{L}(B))$, is defined as $\sqrt{\det(B^T B)}$. It is easy to see when B is full-rank ($n = m$), its determinant becomes $|\det(B)|$.

Two lattices $\mathcal{L}(B_1)$ and $\mathcal{L}(B_2)$ are exactly the same when there exists a matrix $U \in GL_n(\mathbb{Z})$ s.t. $B_1 = UB_2$. Lattices used in cryptography are usually "integer lattices" whose basis matrices are over $\mathbb{Z}$ instead of $\mathbb{R}$. Thus the space of all full-rank integer lattices is actually $(GL_n(\mathbb{R}) \cap \mathbb{Z}^{n\times n})/GL_n(\mathbb{Z})$.

Hermite Normal Form(HNF) is a useful tool to study integer matrices:

Definition 1. *A square nonsingular integer matrix $H \in \mathbb{Z}^{n\times n}$ is in Hermite Normal Form(HNF) if*

- *H is upper triangular, i.e., $h_{ij} = 0$ for all $i > j$.*
- *All diagonal elements are positive, i.e., $h_{ii} > 0$ for all i.*
- *All non diagonal elements are reduced modulo the corresponding diagonal element at the same column, i.e., $0 \le h_{ij} < h_{jj}$ for all $i < j$.*

And there exists a famous result for HNF[5]:

Theorem 1. *For every $A \in GL_n(\mathbb{R}) \cap \mathbb{Z}^{n\times n}$, there exists a unique $n \times n$ matrix $B \in S_{n,\mathbb{Z}}$(HNF) of the form $B = UA$ with $U \in GL_n(\mathbb{Z})$.*

By this theorem, an integer lattice corresponds to its unique HNF, implying that generating an integer lattice is actually equivalent to generating a HNF.

3 Random integer lattice

3.1 Definition

In this part we refer to [21] to recall some results related to random integer lattice.

First, for $M, N \in \mathbb{Z}^+$,

$$H_n^{\le}(M) \triangleq \{H \text{ is } n\text{-dim HNF} | \det(H) \le M\},$$

$$H_n(N) \triangleq \{H \text{ is } n\text{-dim HNF} | \det(H) = N\}.$$

Gruber [9] counted the size of $|H_n(N)|$:

Theorem 2. *If N has prime decomposition $N = p_1^{r_1} \dots p_t^{r_t}$, then*

$$|H_n(N)| = \prod_{i=1}^{t} \prod_{j=1}^{n-1} \frac{p_i^{r_i+j} - 1}{p_i^j - 1}.$$

And there exists an asymptotic estimation for $|H_n^{\le}(M)|$ in [21]:

Theorem 3. *For large positive integer M,*

$$|H_n^{\le}(M)| = \frac{\prod_{s=2}^{n} \zeta(s)}{n} M^n + O(M^{n-1} \log M).$$

H is called an n-dim random nonsingular HNF if for large integer $M > 0$, H is chosen from $H_n^{\le}(M)$ **uniformly**, and the lattice $\mathcal{L}(H)$ generated by such H is called an random integer lattice.

3.2 Diagonal Distribution

In [21], G.Hu et al. studied the expectation and variance of every entry and the probability distribution of every diagonal entry:

Theorem 4. *Let $H = (h_{ij})$ be an n-dim random nonsingular HNF with determinant bounded by $M > 0$ and t be an integer in $[1, n-1]$, given an increasing subset $\{i_1, \cdots, i_t\}$ of $\{1, \cdots, n\}$ and its increasing complementary subset $\{j_1, \cdots, j_{n-t}\}$, for positive integers $b_1 \cdots b_t$, when $M \to +\infty$, we have*

$$P(h_{i_k,i_k} = b_k \text{ for all } k) = \begin{cases} 0 & (i_t = n) \\ \frac{\prod_{k=1}^{n-t-1} \zeta(n+1-j_k)}{\prod_{s=2}^{n} \zeta(s)} \prod_{l=1}^{t} b_l^{i_l - n - 1} & (i_t < n) \end{cases} \tag{1}$$

If we take $t = 1$, a one-element set $T = \{i\}(i \in [1, n-1])$ and positive integers b, then the increasing complementary subset of T in $\{1, 2, \cdots, n\}$ is $\{1, \cdots, i-1, i+1, \cdots, n\}$. We apply the above theorem and obtain the following corollary:

Corollary 1. *Let $H = (h_{ij})$ be an n-dim random nonsingular HNF with determinant bounded by $M > 0$, then for $i \in [1, n-1]$ and positive integer b, when $M \to +\infty$, we have*

$$P(h_{ii} = b) = \frac{1}{\zeta(n+1-i) \cdot b^{n+1-i}}. \tag{2}$$

We denote this distribution of h_{ii} by $\mathbb{D}(n, i)$.

4 Generate Random Integer Lattice via HNF

In this section we present our random integer lattice generation algorithm via HNF. Firstly, we introduce the inverse sampling method in probability theory to generate all the diagonal elements. Then we generate all the non-diagonal elements accordingly.

4.1 Inverse Sampling Method

Given a distribution $\mathbb{D}$ over some ordered set A, we can use inverse sampling method to generate a random variable according to the distribution $\mathbb{D}$. We present the two versions of inverse sampling method: continuous-ISM and discrete-ISM.

Theorem 5. (continuous-ISM) *For distribution $\mathbb{D}$ over interval $[a, b]$ with cumulative distribution function $F_X(x)$, choose a random y uniformly from $[0, 1]$ and compute z s.t. $F(z) = y$, then the resulting variable Z has distribution $\mathbb{D}$.*

Proof. Our goal is to prove Z has F_X as its cumulative distribution function. Namely, for any $x \in [a, b]$, we have to prove $P(Z \leq x) = F_X(x)$. Since F is a monotonically increasing function, we have

$$P(Z \leq x) = P(F_Z(z) \leq F_X(x)) = P(y \leq F_X(x)) = F_X(x)$$

where the second equality comes from $F(z) = y$ and the last one is a direct result of $y's$ uniformity in $[0, 1]$. Thus the cumulative distribution function of Z is actually F_X, which completes the proof.

Theorem 6. (discrete-ISM) *For distribution $\mathbb{D}$ over finite ordered set $A = \{a_k\}_{k=1}^n \subseteq \mathbb{Z}$ with corresponding density $f_k = P(X = a_k)$, choose a random number y uniformly from $[0, 1]$ and compute the minimum j s.t. $\sum_{k=1}^{j} f_k \geq y$, then we let $Z = a_j$ and Z will have distribution $\mathbb{D}$.*

Proof. For any $a_j \in A$, we need to prove $P(Z = a_j) = f_j$. Since j is the minimum value s.t. $\sum_{k=1}^{j} f_k \geq y$, we know that $\sum_{k=1}^{j-1} f_k < y$. Then we have

$$\begin{aligned} P(Z = a_j) &= P(\sum_{k=1}^{j} f_i \geq y, \sum_{k=1}^{j-1} f_i < y) \\ &= P(\sum_{k=1}^{j} f_k \geq y) - P(\sum_{k=1}^{j-1} f_k \geq y)) \\ &= \sum_{k=1}^{j} f_k - \sum_{k=1}^{j-1} f_k \text{ (since } y \text{ is uniform in } [0, 1]) \\ &= f_j \end{aligned}$$

which completes the proof.

4.2 Generate Random Integer Lattice via HNF

From Section 2.2, we can generate a random integer lattice by equivalently generating a random nonsingular HNF. To begin with, we generate the first $n - 1$ diagonal elements $h_{11}, h_{22}, \cdots, h_{n-1,n-1}$. Then we generate the last diagonal element h_{nn}. Finally, all the non-diagonal elements are generated and we output the matrix H as a lattice basis for our random integer lattice.

Generate $h_{11}, h_{22}, \cdots, h_{n-1,n-1}$ From Corollary 1, we know that for n-dim nonsingular HNF, when $i \in [1, n-1]$, the distribution of h_{ii} is

$$\mathbb{D}(n, i) : P(X = x) = \frac{1}{\zeta(n + 1 - i)} \cdot x^{-(n+1-i)} \ (x = 1, 2 \dots).$$

So we generate these diagonal elements $h_{11}, h_{22}, \cdots, h_{n-1,n-1}$ according to $\mathbb{D}(n, i)$ by discrete-ISM(Theorem 6).

For $i \in [1, n-1]$, we choose y uniformly random from [0, 1] and increasingly iterate j_i starting from 1 until it satisfies $\frac{1}{\zeta(n+1-i)} \sum_{k=1}^{j_i} k^{-(n+1-i)} \geq y$. Then we set $h_{ii} = j_i$. By Theorem 6, each diagonal h_{ii} has distribution $\mathbb{D}(n, i)$ as what we need.

Generate h_{nn} After generating the first $n-1$ diagonal elements h_{ii}, we set $D_{n-1} := \prod_{i=1}^{n-1} h_{ii}$. Since the determinant upper bound is M, the last diagonal element h_{nn} should be in $[1, \lfloor \frac{M}{D_{n-1}} \rfloor]$. We point out that D_{n-1} is a small number with high probability, thus $\lfloor \frac{M}{D_{n-1}} \rfloor$ is still large enough for us to obtain a similar result for h_{nn}. For fixed first $n-1$ diagonal elements $h_{11}, \cdots, h_{n-1,n-1}$ and the last diagonal element $h_{nn} = N$, we know that the exact number of such HNFs is $N^{n-1} \cdot \prod_{i=1}^{n-1} h_{ii}^{i-1}$. Thus, for random nonsingular HNF with determinant bounded by M, on the condition that $\prod_{i=1}^{n-1} h_{ii} = D_{n-1}$, the distribution of h_{nn} is the following

$$\begin{aligned}
\tilde{\mathbb{D}}(n, M, D_{n-1}) : & P(X = x) \\
&= \frac{1}{\sum_{k=1}^{\lfloor M/D_{n-1} \rfloor} k^{n-1}} \cdot x^{n-1} \\
&= \frac{1}{\frac{1}{n} \lfloor \frac{M}{D_{n-1}} \rfloor^n + O(\lfloor \frac{M}{D_{n-1}} \rfloor^{n-1})} \cdot x^{n-1} \ (x = 1, 2 \ldots, \lfloor \frac{M}{D_{n-1}} \rfloor).
\end{aligned}$$

Moreover, the corresponding cumulative distribution function is

$$\begin{aligned}
F_X(x) &= P(X \leq x) \\
&= \frac{1}{\frac{1}{n} \lfloor \frac{M}{D_{n-1}} \rfloor^n + O(\lfloor \frac{M}{D_{n-1}} \rfloor^{n-1})} \cdot \sum_{k=1}^{x} k^{n-1} \\
&= \frac{\frac{1}{n} x^n + O(x^{n-1})}{\frac{1}{n} \lfloor \frac{M}{D_{n-1}} \rfloor^n + O(\lfloor \frac{M}{D_{n-1}} \rfloor^{n-1})} \ (x = 1, 2 \ldots, \lfloor \frac{M}{D_{n-1}} \rfloor).
\end{aligned} \tag{3}$$

Since $\lfloor \frac{M}{D_{n-1}} \rfloor$ is still super large, we know

$$F_X(x) \approx \frac{x^n/n}{\lfloor M/D_{n-1} \rfloor^n/n} = (\frac{x}{\lfloor M/D_{n-1} \rfloor})^n$$

is a pretty good estimation for $F_X(x)$.

In fact, let $\tilde{\mathbb{D}}(n, M, D_{n-1})$ defined as above and $\tilde{\mathbb{D}}_0(n, M, D_{n-1})$ be the distribution defined by cumulative distribution function $G_X(x) = (\frac{x}{\lfloor M/D_{n-1} \rfloor})^n$, we have the following theorem.

Theorem 7. *For $M, D_{n-1} > 0$, the statistical distance between $\tilde{\mathbb{D}}(n, M, D_{n-1})$ and $\tilde{\mathbb{D}}_0(n, M, D_{n-1})$ is at most $n \cdot O(\frac{D_{n-1}}{M})$.*

Proof. According to 3, the cumulative distribution function of $\tilde{\mathbb{D}}(n, M, D_{n-1})$ is $F_X(x) = \frac{\frac{1}{n}x^n + O(x^{n-1})}{\frac{1}{n}\lfloor \frac{M}{D_{n-1}} \rfloor^n + O(\lfloor \frac{M}{D_{n-1}} \rfloor^{n-1})}$, since $\tilde{\mathbb{D}}_0(n, M, D_{n-1})$'s cumulative distribution function is $G_X(x) = (\frac{x}{\lfloor M/D_{n-1} \rfloor})^n$, denote $\lfloor \frac{M}{D_{n-1}} \rfloor$ by $\tilde{M}$, then $x \leq \tilde{M}$ and for every $x \in [1, \tilde{M}]$, we have

$$
\begin{aligned}
& |F_X(x) - G_X(x)| \\
=& |\frac{\frac{1}{n}x^n + O(x^{n-1})}{\frac{1}{n}\tilde{M}^n + O(\tilde{M}^{n-1})} - (\frac{x}{\tilde{M}})^n| \\
=& |\frac{x^n + n \cdot O(x^{n-1})}{\tilde{M}^n + n \cdot O(\tilde{M}^{n-1})} - (\frac{x}{\tilde{M}})^n| \\
=& |\frac{(x^n + n \cdot O(x^{n-1}))\tilde{M}^n - (\tilde{M}^n + n \cdot O(\tilde{M}^{n-1}))x^n}{\tilde{M}^{2n} + n \cdot O(\tilde{M}^{2n-1})}| \\
=& |\frac{n \cdot O(x^{n-1})\tilde{M}^n - n \cdot O(\tilde{M}^{n-1})x^n}{\tilde{M}^{2n} + n \cdot O(\tilde{M}^{2n-1})}| \\
=& |\frac{n \cdot O(\tilde{M}^{n-1})\tilde{M}^n - n \cdot O(\tilde{M}^{n-1})\tilde{M}^n}{\tilde{M}^{2n} + n \cdot O(\tilde{M}^{2n-1})}| (\text{since } x \leq \tilde{M}) \\
=& |\frac{n \cdot O(\tilde{M}^{2n-1})}{\tilde{M}^{2n} + n \cdot O(\tilde{M}^{2n-1})}| \\
=& |\frac{n \cdot O(\frac{1}{\tilde{M}})}{1 + n \cdot O(\frac{1}{\tilde{M}})}| = n \cdot O(\frac{1}{\tilde{M}}) = n \cdot O(\frac{D_{n-1}}{M})
\end{aligned}
$$

which implies that the statistical distance $\tilde{\mathbb{D}}(n, M, D_{n-1})$ and $\tilde{\mathbb{D}}_0(n, M, D_{n-1})$ is bounded by $n \cdot O(\frac{D_{n-1}}{M})$.

Since $\lfloor M/D_{n-1} \rfloor$ is still super large, we can generate h_{nn} according to $\tilde{\mathbb{D}}_0(n, M, D_{n-1})$(close to $\tilde{\mathbb{D}}(n, M, D_{n-1})$) by continuous-ISM(Theorem 5).

We choose y uniformly random from $[0, 1]$ and compute $z \in \mathbb{R}^+$ s.t.

$$(\frac{z}{\lfloor M/D_{n-1} \rfloor})^n = y.$$

Then we set $h_{nn} = \lfloor z \rceil$; By Theorem 6 and Theorem 7, the diagonal h_{nn} has distribution $\tilde{\mathbb{D}}_0(n, M, D_{n-1})$, which is close enough to $\tilde{\mathbb{D}}(n, M, D_{n-1})$.

Generate $h_{ij} (i \neq j)$ This part is relatively easier. For $i, j = 1, \ldots, n$, let h_{ij} be chosen from $[0, h_{jj})$ uniform randomly if $i < j$ and let $h_{ij} = 0$ if $i > j$.

Correctness By the discussion above, for large enough $M > 0$, the distribution of the diagonal $h_{11}, \cdots, h_{nn}$ generated by this algorithm are close enough to its distribution as random nonsingular HNF. For $i < j \in [1, n]$, since a random nonsingular HNF's h_{ij} is uniform in $[0, h_{jj})$ and h_{ij} is generated in the same way, we know the output of this algorithm is also close enough to a real random nonsingular HNF, which implies the correctness of this algorithm.

4.3 Algorithm 1: Generate Random Integer Lattice

Algorithm 1 Random Integer Lattice Generation

Input: Dimension n, large integer M
Output: n-dim random integer lattice $\mathcal{L}$ with $\det(\mathcal{L}) \leq M$

Step 1: Generate $h_{11}, \cdots, h_{n-1,n-1}$
$D_0 = 1$
for $i = 1$ to $n-1$ **do**
  $j_i = 1$
  $s_i = 1$
  choose $y_i \in [0, 1]$ uniformly
  while $s_i < \zeta(n+1-i) \cdot y_i$ **do**
    $j_i = j_i + 1$
    $s_i = s_i + j_i^{-(n+1-i)}$
  end while
  $D_i = D_{i-1} \cdot j_i$
  set $\mathbf{h_{ii}} = \mathbf{j_i}$
end for
Step 2: Generate h_{nn}
choose $y \in [0, 1]$ uniformly
$z = y^{1/n}$
$z = z \cdot \lfloor \frac{M}{D_{n-1}} \rfloor$
set $\mathbf{h_{nn}} = \lfloor \mathbf{z} \rceil$
Step 3: Generate $h_{ij} (i \neq j)$
for $j = 1$ to n **do**
  for $i = 1$ to $j-1$ **do**
    choose $\mathbf{h_{ij}} \in [\mathbf{0}, \mathbf{h_{jj}})$ uniformly
  end for
  for $i = j+1$ to n **do**
    set $\mathbf{h_{ij}} = \mathbf{0}$
  end for
end for
Step 4: Set $\mathbf{H} = (\mathbf{h_{ij}})$ and output $\mathcal{L}(H)$

4.4 Time Complexity of Algorithm 1

Now we analyze the time complexity of Algorithm 1. Obviously, the most time-consuming part of Algorithm 1 is the floating-point operations $s_i = s_i + j_i^{-(n+1-i)}$ inside the while iteration for each i in step 1. Denote the number of computing $s_i = s_i + j_i^{-(n+1-i)}$ in the i-th while iteration by $T(i)$. Notice that

$$P(h_{ii} = 1) = \frac{1}{\zeta(n+1-i)},$$

since $\zeta(s)$ converges to 1 quite fast as s grows, the majority of h_{ii} will be set to 1. In fact, by numerical results, we have following result:

Fact 1: For any integer $n \geq 10$,

$$\frac{1}{\prod_{s=10}^{n} \zeta(s)} \geq 0.999.$$

By this fact, for $i \leq n-10$, all the h_{ii} is very likely to be set to 1, implying that $T(1), T(2), \cdots, T(n-10) = 0$ with probability ≥ 0.999. Then we consider the $T(n-9), T(n-8), \cdots, T(n-1)$. If we set the probability bound for each $T(i)$ to be 0.999, then by accurate numerical results, we have the following table:

T(i)	upper bound
$T(n-9)$	0
$T(n-8)$	1
$T(n-7)$	1
$T(n-6)$	1
$T(n-5)$	2
$T(n-4)$	3
$T(n-3)$	6
$T(n-2)$	19
$T(n-1)$	607

Thus we have the following theorem:

Theorem 8. *The number of floating-point operations performed in Algorithm 1 is bounded by* 1300 *with probability* ≥ 0.99.

Proof. By the above table, $\sum_{i=n-9}^{n-1} T(i)$ is bounded by 640 with probability $\geq 0.999^9$. Since $T(1), T(2), \cdots, T(n-10) = 0$ with probability ≥ 0.999, we know $\sum_{i=1}^{n-1} T(i)$ is bounded by 640 with probability $\geq 0.999^{10} \geq 0.99$. Notice that each $s_i = s_i + j_i^{-(n+1-i)}$ needs two floating-point operations and it also needs another 4 floating-point operations to generate h_{nn} in Step 2, thus with probability ≥ 0.99, the total number of floating-point operations performed in Algorithm 1 is bounded by $640 \cdot 2 + 4 = 1284 < 1300$, which completes the proof.

Remark 1. We point out that the accuracy of floating-point affects the actual running time of Algorithm 1. By experiments, 150 bit is a suitable option.

It is not hard to see that in Algorithm 1, besides the floating-point operations, the rest part of step 1, step 2, step 3 takes $O(n^2), O(1), O(n^2)$ operations respectively. Combining this with Theorem 8, we have the following result:

Theorem 9. *Algorithm 1 outputs a random integer lattice within* $O(n^2)$ *operations with probability* ≥ 0.99.

5 Conclusion

In this paper, we present an improved algorithm for generating random integer lattice and discuss its time complexity. We prove that with probability ≥ 0.99,

this algorithm outputs an n-dim random integer lattice within $O(n^2)$ operations. We point out that there is still space for the improvement of our algorithm and leave it as an open problem.

Acknowledgement. We thank Dr. Yanbin Pan for his wonderful suggestions about this paper and we thank the anonymous referees for putting forward their excellent advice on how to improve the presentation of this paper.

Bibliography

[1] M. Ajtai. Gennerating hard instances of lattice problems. In: STOC 1996, pp. 99-108. ACM Press, New York, 1996.

[2] M. Ajtai, C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence, In: STOC 1997, pp. 284-293. ACM Press, New York, 1997.

[3] T. M. Apostol. Introduction to analytical number theory, Undergraduates Texts in Mathematics, Springer, 1976.

[4] Yuanmi Chen, Phong Q. Nguyen. BKZ 2.0: Better Lattice Security Estimates. In: Asiacrypt 2011, LNCS, vol. 7073, pp. 1-20, volume 7073, Springer, Heidelberg, 2011.

[5] H. Cohen. A Course in Computational Algebraic Number Theory, Graduate Texts in Mathematics, Vol.138, Springer, 1993.

[6] P. D. Domich. Residual Hermite normal form computations, ACM Trans. Math. Software 15(3), 275-286, 1989.

[7] P. D. Domich, R. Kannan, L. E. Trotter. Hermite normal form computation using modulo determinant arithmetic, Mathematics of Operations Research 12(1), 50-59, 1987.

[8] M. A. Frumkin. Complexity question in number theory, J. Soviet Math., 29(29), 1502-1517, 1985.

[9] B. Gruber. Alternative formulae for the number of sublattices, Acta Cryst. A53, 807-808, 1997.

[10] C. Gentry, C. Peikert, V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In: STOC 2008, pp. 197-206. ACM Press, New York, 2008.

[11] J. Hoffstein , J. Pipher, J. H. Silverman. NTRU: A Ring-Based Public Key Cryptosystem, In: Buhler, J.P. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 267-288. Springer, Heidelberg, 1998.

[12] M. S. Hung, W. O. Rom. An application of the Hermite normal form in integer programming. Linear Algebra and its Applications 140, 163-179, 1990.

[13] A. K. Lenstra, H. W. Jr. Lenstra, L. Lovasz. Factoring polynomials with rational coefficients. Mathematische Annalen, 261(4): 513-534, 1982.

[14] C. P. Schnorr, M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. Mathematical Programming 66, 181-199, 1994.

[15] D. Micciancio. Improving lattice based cryptosystems using the Hermite normal form. In Cryptography and Lattices Conference 2001, LNCS, Springer-Verlag, 2001.

[16] D. Micciancio, B. Warinschi. A linear space algorithm for computing the Hermite normal form. In ISSAC 2001, 231-236, 2001.

[17] G. Maze. Natural density distribution of Hermite normal forms of integer matrices, Journal of Number Theory, Vol.131, 2398-2408, 2011.

[18] P.Q. Nguyen, D. Stehle. LLL on the average. In ANTS 2006. LNCS, vol. 4076, pp. 238C256. Springer, Heidelberg, 2006.
[19] C. Pernet, W. Stein. Fast computation of Hermite normal forms of random integer matrices, Journal of Number Theory, Vol.130, 1675-1683, 2010.
[20] O. Regev. On lattices, learning with errors, random linear codes, and cryptography, In: STOC 2005, pp. 84-93. ACM Press, New York, 2005.
[21] Gengran Hu, Yanbin Pan, Renzhang Liu, Yuyun Chen. On Random Nonsingular Hermite Normal Form, Journal of Number Theory, vol.164, pp. 66-86, 2016.
[22] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput., 26(5):1484-1509, 1997.