Journal of Math-for-Industry, Vol. 3 (2011B-2), pp. 99–106

An improvement of key generation algorithm for Gentry's homomorphic encryption scheme from ideal lattices

Naoki Ogura, Go Yamamoto, Tetsutaro Kobayashi and Shigenori Uchiyama

Received on August 7, 2011

Abstract. One way of improving efficiency of Gentry's fully homomorphic encryption from ideal lattices is controlling the number of operations, but our recollection is that any scheme which controls the bound has not proposed. In this paper, we propose a key generation algorithm for Gentry's scheme that controls the bound of the circuit depth by using the relation between the circuit depth and the eigenvalues of a basis of a lattice. We present experimental results that show that the proposed algorithm is practical. We discuss security of the basis of the lattices generated by the algorithm for practical use.

Keywords. homomorphic encryption, ideal lattice, circulant matrix

1. Introduction

Some encryption schemes such as the RSA, Paillier [17], and Okamoto-Uchiyama [16] schemes have a homomorphic property. The homomorphic property provides a feature which enables us to deal with encrypted data without being able to decrypt the data. This property has various applications such as to secure voting systems or cross table generation. Many homomorphic encryption schemes incorporate the homomorphic property for only one operation, i.e., no encryption scheme is capable of evaluating any function. Constructing a fully homomorphic encryption scheme that could evaluate all functions is an important open problem in cryptography that has persisted for many years. In 2009, Gentry [6] solved this problem by using ideal lattices. Gentry showed that a fully homomorphic encryption scheme can be constructed in three stages: First, he proposed an abstract construction of homomorphic encryption schemes for some functions. Second, he embodied the idea with ideal lattices. We call this scheme Gentry's basic scheme. Third, he proposed how to extend the scheme so that it has a fully homomorphic property. We call this scheme Gentry's full scheme.

Here, we concentrate on the basic scheme. This is because the efficiency of the full scheme is much lower than that of the basic scheme. We consider that we can construct a practical full scheme by improving the basic scheme.

The key generation algorithm of Gentry's basic scheme generates random basis of ideal lattices as the private key. A bound for the number of operations depends on these basis. Then, it is difficult to handle the number of executable operations in advance. Therefore, we must repeat the key generation until the scheme can handle the desired number of operations. In other words, controlling the bound enables us to construct efficient Gentry's scheme. Then, the problem naturally arises regarding how to handle the number of operations before generating the keys.

In this paper, we address this problem by proposing a key generation algorithm that controls the bound of the circuit depth by using the relation between the circuit depth and the eigenvalues of a basis of a lattice. That is, the proposed key generation algorithm enables us to create a practical homomorphic encryption scheme for a given number of operations. We discuss security of the basis of the lattices generated by the algorithm for practical use. Also, we describe an efficient implementation of Gentry's scheme and show that the proposed algorithm is practical based on experimental results.

Note that an extended abstract of this paper appears in Proceedings of IWSEC2010 Lecture Notes in Computer Science Vol. 6432, Springer-Verlag, 2010 [15]. This is the full version.

This paper is organized as follows. In Section 2, we briefly describe the ideal lattices and Gentry's scheme. In Section 3, we discuss the problem that is dealt with in this paper. In Section 4, we propose an algorithm to address the problem. In Section 5, we explain the efficiency and the security analysis of the proposed algorithm. In Section 6, we present our conclusions.

2. Preliminaries

In this section, we explain some basic definitions and facts.

2.1. Definitions on lattices

Gentry [6] used ideal lattices for constructing a homomorphic encryption scheme. In this section, we briefly review ideal lattices.

Definition 1 (Ideal Lattices)**.** Let R be a residue class ring of the integer univariate polynomial ring $\mathbb{Z}[x]$ modulo the ideal $(f(x))$, where $f(x)$ is a monic integer univariate polynomial with degree n. Then, R is isomorphic to $\mathbb{Z}^n$ as a $\mathbb{Z}$-module. We define an ideal lattice (on f) as a sublattice of $\mathbb{Z}^n$ isomorphic to an ideal of R.

This isomorphism enables us to introduce multiplication over $\mathbb{Z}^n$ by using that over R. So ideal lattices have two operations: addition as a sublattice of $\mathbb{Z}^n$ and multiplication corresponding to polynomial multiplication modulo f.

One of the most simple ideals of R is a principal ideal. Sublattices corresponding to principal ideals fulfill important roles in constructing practical encryption schemes.

Definition 2 (Rotation Basis)**.**
For vector $v = (v_0,\ v_1, \ldots, v_{n-1})^t \in \mathbb{Z}^n$, we define $\bar{v} := v_0 + v_1 x + \cdots + v_{n-1}x^{n-1} \mod f$ in R. Any element of principal ideal $(\bar{v})$ can be written as a linear combination of generators $\bar{v},\ \bar{v}x, \cdots, \bar{v}x^{n-1}$. By $\mathrm{rot}(v)$, we denote a matrix consisting of these generators.[1]

For example, if $f(x) = x^n - 1$, $\mathrm{rot}(v)$ is the circulant matrix as:

$$\begin{pmatrix} v_0 & v_{n-1} & \cdots & v_2 & v_1 \\ v_1 & v_0 & \cdots & v_3 & v_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ v_{n-2} & v_{n-3} & \cdots & v_0 & v_{n-1} \\ v_{n-1} & v_{n-2} & \cdots & v_1 & v_0 \end{pmatrix}.$$

We refer to the lattice corresponding to the basis as the cyclic lattice.

We can see for $f(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_0$, $\mathrm{rot}(v) = (b_{ij})_{i,j}$ satisfies the following recurring formula,

$$b_{ij} = \begin{cases} v_{i-1} & (1 \le i \le n,\ j = 1) \\ -b_{n\ j-1}a_0 & (i = 1,\ 2 \le j \le n) \\ b_{i-1\ j-1} - b_{n\ j-1}a_{i-1} & (2 \le i \le n,\ 2 \le j \le n) \end{cases}.$$

Definition 3 (Half-Open Parallelepiped)**.** Let $\mathcal{L}$ be a sublattice of $\mathbb{Z}^n$, regardless of whether or not it is an ideal lattice. There are some linear independent vectors $b_1,\ b_2, \cdots, b_m$ of $\mathcal{L}$ such that all elements of $\mathcal{L}$ can be written as linear combinations of these vectors. We define a basis as the $n \times m$-matrix $B := (b_1\ b_2\ \cdots\ b_m)$.[2] For basis $B = (b_1\ b_2\ \cdots\ b_m)$, we define half-open parallelepiped $P(B) := \{\sum_{i=1}^{m} x_i b_i \mid -\frac{1}{2} \le x_i < \frac{1}{2}\}$.

Note that a basis is not uniquely defined for a lattice and so an infinite number of half-open parallelepipeds exist for a specific ideal lattice.

We define the modulo operation by using a half-open parallelepiped.

Definition 4 (Modulo Operation by a Lattice)**.**
Let $\mathcal{L}(B)$ be a lattice with basis B. For vector $t \in \mathbb{Z}^n$, we can find a unique vector t' that satisfies the following conditions:

- t' is equivalent to t: $t - t' \in \mathcal{L}(B)$
- t' is a reduced vector: $t' \in P(B)$

We refer to t' as the remainder of t by B. It is written as $t' \equiv t \pmod{B}$.

We can compute $t \mod B$ as

$$t \mod B = t - B \cdot \lfloor B^{-1}t \rceil ,$$

where for $v \in \mathbb{R}^n$, $\lfloor v \rceil$ is a vector of $\mathbb{Z}^n$ after each element of v is rounded to an integer.

2.2. Gentry's scheme

In [6], a homomorphic encryption scheme over an abstract ring is discussed, and then ideal lattices are proposed as a realization of the ring. In this subsection, we explain Gentry's basic scheme, which has a bound for the circuit depth. We concentrate on the basic scheme since we believe that progress in the basic scheme will lead us to improve the full scheme.

First, we select monic integer polynomial $f(x) \in \mathbb{Z}[x]$ of degree n. Then, we set residue ring $R = \mathbb{Z}[x]/(f(x))$. Also, let B_I be a basis for some ideal $I \subset R$ and define plaintext space $\mathcal{P}$ as (a subset of) $P(B_I) \cap \mathbb{Z}^n$. For example, $\mathcal{P} = \{(b_0,\ b_1, \ldots, b_{n-1})^t \mid b_i \in \{0,\ 1\}$ for $i = 0,\ 1, \ldots, n-1\}$ for the scalar diagonal basis $B_I = 2E_n$ corresponding to $I = (2)$, where E_n is the identity matrix of size n. Moreover, we select short vector $s \in \mathcal{L}(B_I)$, where $\mathcal{L}(B_I)$ is the sublattice with basis B_I.[3] For instance, we can use $s = (2,\ 0,\ 0, \ldots, 0)^t$ for $B_I = 2E_n$. For $\phi_1,\ \phi_2 \in \mathbb{Z}^n$, we define $\phi_1 +_I \phi_2 := (\phi_1 + \phi_2) \mod B_I$. Similarly, we define $\phi_1 \times_I \phi_2 := (\phi_1 \times \phi_2) \mod B_I$, $\phi_1 +_J \phi_2 := (\phi_1 + \phi_2) \mod B_J^{\mathrm{pk}}$, and so on.

[KeyGen]
Generate two basic matrices B_J^{pk} and B_J^{sk} corresponding to ideal J relatively prime to I. Then, the public-key is B_J^{pk} and the secret-key is B_J^{sk}. Typically, we can use B_J^{sk} as $\mathrm{rot}(v)$ for random vector v with the corresponding polynomial prime to I. Also, we may set B_J^{pk} as the Hermite normal form[4] of B_J^{sk}. We propose a more concrete key generation algorithm for improving the homomorphic property later.

[Encrypt]
For a plaintext π, output $\phi := (\pi + r \times s) \mod B_J^{\mathrm{pk}}$, where $r \in \mathbb{Z}^n$ is chosen randomly such that $\|r\| \le \ell$. Note that ℓ is a security parameter that we determine later.

[Decrypt]
For a ciphertext ϕ, output $\pi := (\phi \mod B_J^{\mathrm{sk}}) \mod B_I$.

[Evaluate]
For circuit C_I and tuple $(\phi_1, \ldots, \phi_t)$ of ciphertexts, output $C_J(\phi_1, \ldots, \phi_t)$, where C_J is the circuit replaced by C_I using gate $+_J,\ \times_J$ instead of gate $+_I,\ \times_I$.

[1] Gentry[6] refers to such a basis a "rotation basis."

[2] The terminology "basis" is typically defined as not a matrix but a set of vectors. In this paper, we follows Gentry's notation about it.

[3] As described in [6], we can also select s randomly for every encryption. In the current situation, we select s in advance to improve the homomorphic property.

[4] The Hermite normal form for a lattice is a unique basis and can be efficiently computed. See [14] for more information.

Gentry discussed the validity of Evaluate. See [6] and [7] for more information.

Definition 5 (ρ_{Enc})**.** ρ_{Enc} is value

$$\rho_{\mathrm{Enc}} := \max_{\pi \in \mathcal{P},\ \|r\| \le l} \|\pi + r \times s\| .$$

For example, $\rho_{\mathrm{Enc}} \le \sqrt{n} + 2\ell$ for $I = (2)$, $s = (2,\ 0,\ 0, \cdots, 0)^t$. In this paper, we use ℓ satisfying $\rho_{\mathrm{Enc}} \le n$ for the sake of simplicity.

The following value expresses the size of $P(B)$.

Definition 6 (ρ_{Dec})**.** ρ_{Dec} is value

$$\rho_{\mathrm{Dec}} := \sup\{\rho \in \mathbb{R}_{>0} \mid \mathcal{B}_\rho \subset P(B)\} ,$$

where $\mathcal{B}_\rho := \{t \in \mathbb{R}^n \mid \|t\| < \rho\}$.

In fact, ρ_{Dec} can be determined by basis B_J^{pk}. In what follows, we set $B = B_J^{\mathrm{pk}}$ for simplicity.

Lemma 1 ([6, Lemma 1])**.**
For $(b_1{}^\star\ b_2{}^\star\ \cdots\ b_n{}^\star) := (B^{-1})^t$,

$$\rho_{\mathrm{Dec}} = \frac{1}{2\max_j \|b_j{}^\star\|} .$$

Then, we quote the following important theorem. The theorem states that the bound of the circuit depth depends on value ρ_{Dec}. Note that lg denotes the logarithm function to base 2.

Theorem 1 ([6, Theorem 8])**.**
Set $\gamma := \max\left\{2,\ \sup_{u,\ v \ne 0} \frac{\|u \times v\|}{\|u\|\|v\|}\right\}$.
Assume that the depth of a circuit C *is less than or equal to*

$$\lg \frac{\lg \rho_{\mathrm{Dec}}}{\lg(\gamma \rho_{\mathrm{Enc}})} .$$

Then, Evaluate for C *(and any tuple of ciphertexts) is valid.*

3. Bound of the circuit depth

In this section, we raise some questions that are related to the bound of the circuit depth.

3.1. Reasoning for considering the bound of the circuit depth

Gentry achieved a construction of a bootstrappable scheme by using a server aided cryptographic technique. Roughly speaking, the bootstrappable property is such that we can validly execute Evaluate for the decryption circuit. If we have a bootstrappable scheme, we can construct a homomorphic encryption scheme for any given operation bound by using Gentry's technique.

In this subsection, we discuss the potential to improve Gentry's scheme. As mentioned earlier, the bound of the depth of circuits is connected to ρ_{Dec}, which is determined by the basis of a lattice. If we selected the basis randomly as Gentry suggested, we cannot predict the bound of the circuit depth before generating keys. Then, we must increase the key size or repeat the key generation until the scheme can handle the bound of the circuit depth. Thus, the complexities of encryption/decryption or key generation are increased. Conversely, if we can control the bound of the circuit depth, we can minimize the key size and time-complexity. We may use a homomorphic encryption scheme to construct particular cryptographic protocols where the number of involved parties is bounded. In this case, we can estimate the bound of the circuit depth. Then, the problem naturally arises of how to handle the number of operations before generating the keys. In this paper, we address this problem.

Note that we can construct a homomorphic encryption that has any bound for the circuit depth by using the full scheme. However, the full Gentry scheme requires an additional security requirement to the basic scheme. That is, the full scheme is based on the difficulty of not only the problem corresponding to the basic scheme but also a problem associated with server aided cryptography. Also, since the full scheme is constructed by applying the bootstrapping technique to the basic scheme, the efficiency of the full scheme is much lower than that of the basic scheme. By improving the basic scheme, we can consequently increase the efficiency of the full scheme through a reduction in the number of times the bootstrapping technique is applied. So we concentrate on the basic scheme.

3.2. Circuit depth and eigenvalue

The bound of the circuit depth is connected to ρ_{Dec}, which is determined by the basis of a lattice as shown in Theorem 1. In this subsection, we show that the value is closely related to the eigenvalues of the basis. In what follows, elements of matrices are in the complex field.

At first we define the notion called matrix norms.

Definition 7. Let A be an n-dimensional square matrix. Then, the spectral norm of A is the value

$$\|A\| := \max_{\|x\|=1} \|Ax\| .$$

Also, for $A = (a_{ij})$, the Frobenius norm of A is the value

$$\|A\|_F := \sqrt{\sum_{i,j} |a_{ij}|^2} .$$

As is well known, $\|A\| = \sqrt{\lambda_{|\max|}(A^* A)}$, where A^* is the complex conjugate matrix of the transpose matrix A^t of A. Also, we denote the maximum and minimum of the absolute eigenvalues of A by $\lambda_{|\max|}(A)$ and $\lambda_{|\min|}(A)$, respectively. We can easily see $\|A\| \le \|A\|_F$. Then, we deduce the following theorem from these properties.

Theorem 2. *For a real non-singular matrix* B,

$$\frac{\sqrt{\lambda_{|\min|}(B^* B)}}{2} \le \rho_{\mathrm{Dec}} \le \frac{n\sqrt{\lambda_{|\min|}(B^* B)}}{2} .$$

Proof. We denote column vectors of $(B^{-1})^*$ by $(b_1{}^\star\ b_2{}^\star\ \cdots b_n{}^\star)$. Then,

$$\max_j \|b_j{}^\star\| \le \max_{\|x\|=1} \|(B^{-1})^* x\| = \|(B^{-1})^*\| \ .$$

So we have

$$\begin{aligned}\max_j \|b_j{}^\star\| &\ge \frac{1}{n}\sum_j \|b_j{}^\star\| \\ &\ge \frac{1}{n}\|(B^{-1})^*\|_F \\ &\ge \frac{1}{n}\|(B^{-1})^*\| \ .\end{aligned}$$

Thus, the following equation and Lemma 1 imply the theorem.

$$\begin{aligned}\|(B^{-1})^*\| &= \sqrt{\lambda_{|\max|}(B^{-1}(B^{-1})^*)} \\ &= 1/\sqrt{\lambda_{|\min|}(B^*B)} \ .\end{aligned}$$ □

The theorem says that the bound of the circuit depth is linked to the eigenvalues of B^*B. Also, for $B = (b_{ij})$, we have

$$\max_{i,j} |b_{ij}| \le \|B\| = \sqrt{\lambda_{|\max|}(B^*B)} \ .$$

So the eigenvalues are also involved in the size of each elements of B.

3.3. Handling the eigenvalues

Gentry [6] says that we may generate keys as $\mathrm{rot}(v)$ for some random vector v. So we analyze eigenvalues of $\mathrm{rot}(v)$.

Theorem 3. *Set $B = \mathrm{rot}(v)$ for $v = (v_0,\ v_1, \cdots, v_{n-1})^t$ on $f(x)$ with degree n. We denote all roots (over the field) of $f(x) = 0$ by $\alpha_1,\ \alpha_2, \cdots, \alpha_n$ (counted up to its multiplicity).*

Then, if all roots α_i are distinct, the eigenvalues of B are

$$\lambda_i := \sum_{k=0}^{n-1} v_k \alpha_i{}^k \ ,$$

and B can be diagonalized. More precisely, for $P = (\alpha_i{}^{j-1})_{1\le i,j\le n}$, $PBP^{-1} = \Lambda$, where Λ represents the diagonal matrix each diagonal element $\Lambda_{i,i}$ for which is λ_i.

Proof. For $B = (b_{ij})$, it is only necessary to prove equation

$$\sum_{k=1}^{n} b_{kj}\alpha_i{}^{k-1} = \lambda_i \alpha_i{}^{j-1},$$

for any $1 \le i,\ j \le n$. Note that P is invertible if all α_i's are distinct. The equation can be easily proved by induction on j for any (fixed) i. □

Note that it is not always true that eigenvalues of B^tB can be determined by eigenvalues of real matrix B. However, if $P^t = P$, that is, P is symmetric, then the statement is always true. Especially, if B is a circulant matrix, that is, $f(x) = x^n - 1$, invertible matrix P equals discrete Fourier transformation matrix $W = (\omega^{ij})$, where ω is a primitive n-th root of unity. Then, W is a symmetric matrix.

Note that if $|v_i|$ is bounded by some constant c and $|\alpha_i| \ne 1$, λ_i is bounded as follows.

$$\begin{aligned}|\lambda_i| &= |\sum_{k=0}^{n-1} v_k\alpha_i{}^k| \\ &\le \sum_{k=0}^{n-1} |v_k||\alpha_i|^k \\ &\le c\frac{|\alpha_i|^n - 1}{|\alpha_i| - 1} \ .\end{aligned}$$

This means that c must be large if $|\alpha_i| \sim 1$. Especially, for $f(x) = x^n - 1$, $\lambda_i \sim 0$ in the case that $\alpha_i \ne 1$ and $v_1,\ v_2, \cdots, v_n \sim c$. Thus, it is expected that ρ_{Dec} take a small value if v_i's are generated randomly. We can also generate v_i by selecting vectors that are almost parallel to $e_i := (0,\ 0, \cdots, 0,\ 1,\ 0, \cdots, 0)$. A similar way may also be used in key generation for GGH cryptosystems [9]. In [9], two key generation methods were proposed. One method is to generate keys randomly and the other is to generate values by adding short random vectors to a vector which equals the multiplication of e_i by a large constant. Goldreich et al. comment that attackers may obtain a clue into breaking the scheme if the latter is used.

Note that it is not easy to generate a secure key, i.e. basis, that does not correspond to $\mathrm{rot}(v)$ for some v. This is because ideal lattices have a special construction. Let $\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_k$ be generators of ideal $I \subset R$. Also, we denote the integer vector corresponding to $\bar{v}_i$ by v_i. Then, a basis of the ideal lattice for I should generate the column vectors of $\mathrm{rot}(v_i)$. So the size of the basis would be small compared to the size of v_i.

Thus it would seem that we cannot predict the bound of the circuit depth if we use usual key-generating methods such as random generation. Therefore, we propose another algorithm to address this problem. We approach the problem by controlling the eigenvalues in advance.

4. How to control the circuit depth

In this section, we describe the proposed algorithm.

4.1. Key idea

The proposed strategy for solving the problem is to take a basis where the sizes of the eigenvalues for which are ensured instead of generating keys randomly. However, there is a problem in implementing this strategy: elements of B can be in the complex field. We address this problem by considering each element of B as an element in an integer residue ring in which $f(x)$ can be completely factored.

Here, we describe the main points of the algorithm. First, for circuit depth bound d, we estimate ρ by using Theorem 1. We recall that we assume $\rho_{\mathrm{Enc}} \le n$. Second, we select a suitable m for regarding roots of $f(x)$ as elements of integer residue ring $\mathbb{Z}/m\mathbb{Z}$. We provide an algorithm for selecting m by using a splitting field of $f(x)$ over $\mathbb{Q}$. Third, we select randomly λ_i such that $|\lambda_i|/2 \geq \rho$. If λ_i's are eigenvalues of rotation basis B, the relation between ρ_{Dec} and λ_i shown by Theorem 2 ensures that $\rho_{\mathrm{Dec}} \geq \rho$. That is, the bound of the circuit depth is greater than d. Finally, we have B with the relation between eigenvalues λ_i and B derived using Theorem 3. Note that we can obtain v such that $B = \mathrm{rot}(v)$ by $v = (\mathrm{rot}(v))_1 = B_1 = (P^{-1}\Lambda P)_1 = P^{-1}\Lambda P_1$.

4.2. Proposed algorithm

Here we show key generation algorithm that preserves the homomorphic property for the circuit where the depth is bounded by a given value in Table 1.

Table 1: Key Generation Algorithm for Gentry's Scheme

```
Input: $d$: Bound of the circuit depth,
       $f(x)$: Monic integer univariate polynomial
       such that $n = \deg(f)$
Output: $(B^{\mathrm{pk}},\ B^{\mathrm{sk}})$: the pair of keys for
       Gentry's scheme

1. Compute $\rho := (n\gamma)^{2^d}$ for
   $\gamma := \max\left\{2,\ \sup_{u,\ v\neq 0} \frac{\|u \times v\|}{\|u\|\|v\|}\right\}$.
2. Compute a (not necessarily minimal) splitting
   field $\mathbb{Q}(\theta)$ of $f(x)$ over $\mathbb{Q}$.
3. Compute the minimal polynomial $g(x)$ of $\theta$.
4. Compute $m = |g(i)|$ for randomly generated integer $i$.
5. If the denominator of a root of $f(x)$ over $\mathbb{Q}$ is
   not prime to $m$, then Goto 4.
6. Call the function GenKeyWith$\rho(f(x),\ m,\ \rho)$ and
   output the returned values.
```

Table 2: GenKeyWithρ (function)

```
Input: $f(x)$: Monic polynomial, $m$ and $\rho$: Integers
Output: $(B^{\mathrm{pk}},\ B^{\mathrm{sk}})$: the pair of keys for
       Gentry's scheme

1. Select $\lambda_1,\ \lambda_2, \cdots, \lambda_n$ randomly such that
   $2\rho \leq |\lambda_i| < m$.
2. Construct $P = ({\alpha_i}^{j-1})$ over $\mathbb{Z}/m\mathbb{Z}$,
   where $f(x) = \prod_{i=1}^{n}(x - \alpha_i) \mod m$.
3. Compute $v = P^{-1}\Lambda P_1$, where $P_1$ is
   the first column vector of $P$.
4. Compute $B = \mathrm{rot}(v)$.
5. Output the integer matrix $B^{\mathrm{sk}}$ corresponding to $B$.
6. Compute the Hermite normal form of $B^{\mathrm{sk}}$ and
   output the matrix as $B^{\mathrm{pk}}$.
```

For the selection of m, we execute steps 2 to 5 in Table 1. In our algorithm, we must have all roots α_j of $f(x)$ over $\mathbb{Z}/m\mathbb{Z}$. The following proposition ensures that $f(x)$ splits in $\mathbb{Z}/m\mathbb{Z}$.

Proposition 1. *Let $f(x) \in \mathbb{Z}[x]$ be a monic polynomial of degree n. Also, let $K = \mathbb{Q}(\theta)$ be a number field where θ is a root of a monic irreducible polynomial $g(y) \in \mathbb{Z}[y]$. We define $h_1(y), h_2(y), \ldots, h_n(y) \in \mathbb{Q}[y]$ such that $f(x) = \prod_{j=1}^{n}(x - h_j(\theta))$. Let $d = \prod_{j=1}^{k} d_j$ where the $d_j \in \mathbb{Z}_{>0}$ is chosen so that the $d_j h_j(y)$ have integer coefficients and are primitive. Then, we have the following congruence equation.*

$$d\left(f(x) - \prod_{j=1}^{n}(x - h_j(y))\right) \equiv 0 \pmod{g(y)}$$

Proof. Because of the definition of d, we have

$$d\left(f(x) - \prod_{j=1}^{n}(x - h_j(y))\right) \in \mathbb{Z}[x, y] \ .$$

By using Euclidean division over $(\mathbb{Z}[x])[y]$, there are some $u(x,\ y), v(x,\ y) \in \mathbb{Z}[x,\ y]$ such that

$$d\left(f(x) - \prod_{j=1}^{n}(x - h_j(y))\right) = g(y)u(x,\ y) + v(x,\ y), \quad (1)$$

where $\deg_y u(x,\ y) > \deg_y v(x,\ y)$. By substituting $y = \theta$ for the equation (1), we obtain

$$0 = v(x,\ \theta) \ .$$

The minimality of $g(y)$ implies that all coefficients of $v(x,\ y) \in (\mathbb{Z}[y])[x]$ are divisible by $g(y)$. Thus, we have $v(x,\ y) = 0$ owing to the condition of the degree. Therefore, the equation (1) gives the proposition. □

For $m = |g(i)|$ ($^{\exists}i \in \mathbb{Z}$), Proposition 1 shows that the roots of $(\prod_{j=1}^{k} d_j)f(x)$ are $h_1(i), h_2(i), \ldots, h_n(i)$. If the denominator d_j of $h_j(y)$ is prime to m, $h_j(i)$'s are roots of $f(x)$ over $\mathbb{Z}/m\mathbb{Z}$.

Here we refer to the computation of the splitting field. We may compute the splitting field of $f(x)$ as the following.

1. Factor $f(x)$ over $\mathbb{Q}$. We put $f(x) = \prod_{j=1}^{k} f_j(x)$.

2. Define the number field K_j corresponding to $f_j(x)$, and compute the Galois closure $\bar{K}_i$ of K_i.

3. Compute the compositum K of $\bar{K}_1, \bar{K}_2, \ldots, \bar{K}_k$.

This algorithm works well if the sizes of the related Galois groups are small.

If we know the minimal splitting field of f in advance, we can skip the computation of the splitting field. For example, the splitting field of $f(x) = x^n - 1$ is known as the cyclotomic field, which is generated by a primitive n-th root of unity. The following proposition shows that a primitive root of unity can be expressed over an integer residue ring.

Proposition 2. *Let n be a power of 2. Set $m := \omega^{n/2} + 1$ for a power ω of 2. Then, ω is a primitive n-th root of unity over $\mathbb{Z}/m\mathbb{Z}$.*

Proof. The theorem follows immediately from the following congruent equation.

$$\omega^{n/2} \equiv -1 \pmod{m} . \qquad \square$$

Next, we propose another algorithm in Table 3. Note that we do not input $f(x)$ but n in the algorithm.

Table 3: Another Key Generation Algorithm for Gentry's Scheme

```
Input: $d$: Bound of circuit depth, $n$: Integer
Output: ($B^{\rm pk}$, $B^{\rm sk}$): the pair of keys for
        Gentry's scheme, $f(x)$: Monic integer polynomial
1. Compute $\rho := (n\gamma)^{2^d}$ for
   $\gamma := \max\left\{2, \sup_{u,\ v \neq 0} \frac{\|u \times v\|}{\|u\|\|v\|}\right\}$.
2. Generate randomly $m$ such that $m \geq 2\rho$.
3. Generate integers $\alpha_i \in \mathbb{Z}$ for $i = 1, 2, \ldots, n$.
4. Compute $\tilde{f}(x) = \prod_{i=1}^{n}(x - \alpha_i)$.
5. Compute $f(x)$ such that $f(x) \equiv \tilde{f}(x) \pmod{m}$.
   by adding random multiples of $m$ to each coefficients
   of $\tilde{f}$ except for the term $x^n$.
6. Output $f(x)$.
7. Call the function GenKeyWith$\rho$($f(x)$, $m$, $\rho$) and
   output the returned values.
```

For creating $f(x)$, we execute steps 3 to 6 in Table 3. In this algorithm, we define $f(x)$ for randomly chosen m, instead of selecting m for given $f(x)$.

If we use the second algorithm (Table 3), on the other hand it is not necessary to compute the splitting field of $f(x)$, but encryption or decryption algorithms may be slow, because these algorithms are affected by the form of $f(x)$. So we recommend the first algorithm if the form of $f(x)$ is fixed or selected in a limited way. We should use the second algorithm if the form of $f(x)$ ought to be randomly chosen because, for example, an attacker could find a way of exploiting a security hole related with the form of $f(x)$.

4.3. Feasible bound of the circuit depth

In this subsection, we estimate a feasible bound for the circuit depth. Considering the security requirements, we could not use too large a circuit depth. As mentioned in Section 5.2, the condition that $\sqrt{n}2\rho < 2^{n^{1-\delta}}$ must be satisfied, where $\delta \in [0,\ 1)$ is a security parameter. Thus, we can estimate the maximum circuit depth as follows.

Proposition 3. *Assume that ρ satisfies the condition $\sqrt{n}2\rho < 2^{n^{1-\delta}}$. Then, the bound of circuit depth d is less than*

$$\left\lfloor \lg\left(\frac{n^{1-\delta} - \lg(2\sqrt{n})}{\lg(n\gamma)}\right)\right\rfloor .$$

For example, if $\delta = \frac{1}{8}$, we can construct Gentry's scheme with the circuit depth of 3 for $f(x) = x^{256} - 1$.

5. Analysis of the proposed algorithm

In this section, we analyze the efficiency and the security of the proposed algorithm.

5.1. Practicality of the proposed algorithm

First, we consider $f(x) = x^n - 1$ in terms of efficiency. As noted in Section 3.3, if $f(x) = x^n - 1$, then P is a discrete Fourier transformation matrix. So techniques for fast Fourier transformation can be applied to the algorithm. Since $\Lambda P_1 = (\lambda_1,\ \lambda_2, \cdots, \lambda_n)^t$, we can compute vector $v = P^{-1}\Lambda P_1$ by applying fast Fourier transformation techniques (on $P^{-1} = (\frac{1}{n}\omega^{-ij})$) to $(\lambda_1,\ \lambda_2, \cdots, \lambda_n)^t$. Note that the fast Fourier transformation is efficient if n is a power of 2.

Next, we describe implementation techniques for Gentry's scheme. Since the modulo operation by a lattice is the most time-consuming in Gentry's scheme, we consider how to improve its operation. If we take B_I as scalar matrix $2E_n$, $A = (a_{ij}) \mod B_I$ can be easily computed using $(a_{ij} \mod 2)$. Also, to speed up the encryptions, the inverse matrix of $B_J^{\rm pk}$ is precomputed. Moreover, $B_J^{\rm sk} = {\rm rot}(v)$ can be computable efficiently by using

$${\rm rot}(v) \cdot \lfloor {\rm rot}(v)^{-1}\phi \rceil = v \times \lfloor w \times \phi \rceil ,$$

where $w \in \mathbb{Q}^n$ [5] satisfies $v \times w = (1,\ 0,\ 0, \ldots, 0)^t$. Note that $v_1 \times v_2 = {\rm rot}(v_1)v_2$ for $v_1,\ v_2 \in \mathbb{Z}^n$ and $v_1 \times v_2$ can be computed with a polynomial multiplication. Also, element $\bar{w} \in \mathbb{Z}[x]/(f(x))$ corresponding to w is the inverse in $\mathbb{Q}[x]/(f(x))$ of the element $\bar{v}$ corresponding to v. So w (or $\bar{w}$) is computable by applying the extended Euclidean algorithm to $\bar{v}$ and $f(x)$.

Here, we present the experimental results of Gentry's scheme using the proposed algorithm. Before that, we briefly summarize the key generation algorithm. First we generate integers λ_i's for the given number of operations. Then, we obtain the matrix corresponding to a rotation basis with the eigenvalues of λ_i by executing operations over an integer residue ring.

Table 4 shows the experimental results of Gentry's scheme with the proposed algorithm on $f(x) = x^n - 1$. We used a computer with 2-GHz CPU (AMD Opteron 246), 4 GB memory, and a 160 GB hard disk. Note that we used at most 1 GB memory to execute the program. Magma [23] was used as the software for writing the program. We measured the computation times and the amount of memory used for each step, including key generation, encryption, decryption and d times multiplications of ciphertexts. Note that we show the average run time for the multiplication. The number of iterations is 10. We take the average values except the maximum and minimum for each item.

Comparing the experimental results to those of [8], it appears that the proposed algorithm is not very efficient. We used Magma on the computer with 4 GB of memory, while

[5] The isomorphism between $\mathbb{Z}^n$ and $\mathbb{Z}[x]/(f(x))$ is naturally extended to the isomorphism between $\mathbb{Q}^n$ and $\mathbb{Q}[x]/(f(x))$.

Table 4: Experimental Results for Gentry's Scheme on $f(X) = X^n - 1$

n	64		128		256	
d	1	3	1	3	1	3
Keygen [s]	0.93	1.54	20.12	28.21	416.82	416.48
Encrypt [s]	0.000	0.001	0.007	0.007	0.031	0.029
Decrypt [s]	0.030	0.055	0.38	0.61	7.87	7.83
Multiply [s]	0.001	0.002	0.006	0.008	0.047	0.048
Memory [MB]	9.39	10.11	20.61	20.31	77.87	78.79

Gentry et al. used NTL/GMP libraries on a computer with 24 GB of memory. Based on the current experiments, implementations with C seem to be much faster than those for the Magma implementation. To obtain more accurate results, we must compare the experimental results in the same experimentation environment.

Here, we comment regarding the differences between the proposed algorithm and other related schemes.

5.1.1. Smart and Vercauteren's scheme

In [20], an efficient fully homomorphic encryption scheme is proposed. They use a specific lattice inspired with some prime ideals over an algebraic number field. So their scheme is based on the hardness of a strong problem compared to that for the full Gentry scheme. Also, their experimental results show that their scheme has the homomorphic property for circuits but with a depth that would not be deep enough to enable a fully homomorphic encryption scheme. We expect that since the proposed algorithm uses eigenvalues it can be applied to their scheme.

5.1.2. Stehlé and Steinfeld's scheme

In [21], an efficient fully homomorphic encryption scheme is proposed. They give a security analysis of the Sparse Subset Sum Problem, which is one of the hard problems underlying the security of the full scheme. The analysis leads us to smaller parameter choices. Also they improve the decryption algorithm for the full scheme. In contrast, we concentrate on the basic scheme, and the key generation algorithm in particular. In this way, the proposed algorithm is an improvement to Gentry's scheme regarding this specific part and their algorithm focuses on another part. The proposed algorithm would be applied to generate a basis for their scheme.

5.2. Security analysis of the proposed algorithm

Attackers may break Gentry's scheme with a lattice reduction algorithm by finding short vectors. The following well-known theorem yields a bound for the length of the shortest vector with the determinant of the basis.

Theorem 4 (Minkowski). *Let $\alpha(B)$ be the length of the shortest vector in an n-dimensional full lattice with the basis B. Then,*

$$\alpha(B) < \sqrt{n}\det(B)^{1/n} .$$

Note that $\det(B)$ equals the multiplication of all eigenvalues of B. So we can control $\alpha(B)$ by selecting the eigenvalues. Various lattice reduction algorithms were proposed, for example, in [13] or [19]. The most efficient algorithm was proposed by Ajtai et al. [1]. The algorithm can find a vector of length at most $2^{O(n \lg\lg n/\lg n)}$ times the length of the shortest non-zero vector. Also, Gama and Nguyen [5] provide assessments of the practical hardness of the shortest vector problem based on many experimental results. Especially, they explain why the 334-dimensional NTRU lattices [11] have not been solved. Since the NTRU lattice is an ideal lattice, we recommend using $n > 334$.

We analyze the key generation algorithm assuming that we can compute short vectors with the approximate factor $2^{n^{1-\delta}}$. Because we take the size of eigenvalues as almost 2ρ, the condition that $\sqrt{n}2\rho < 2^{n^{1-\delta}}$ should be satisfied. In fact, if $\alpha(B)/\ell \geq 2^n$, Gentry's scheme is broken. For more information, refer to [7].

Of course, the proposed algorithm generates more specially-configured keys than simple random generation. So the security level would decrease by restricting the keys. Investigating the security is for future work.

6. Conclusion

We proposed an efficient key generation algorithm that controls the bound of the circuit depth by using the relation between the circuit depth and eigenvalues of a basis of a lattice. The key generation algorithm enables us to create a homomorphic encryption scheme for a given number of operations. Also, we described an efficient implementation of Gentry's scheme and showed that the proposed algorithm is practical based on experimental results.

The algorithm is summarized as follows. First we generate eigenvalues for the given number of operations. Then, we obtain the matrix corresponding to a rotation basis by using eigenvalues over an integer residue ring.

Although the experimental results show that the algorithm is practical, the efficiency of the algorithm remains a matter of research. Especially, we should improve the bound of the circuit depth. Improving the quality of the algorithm is for future work. For specific lattices such as cyclic lattices, we continue investigating the security of the scheme with the proposed method.

Acknowledgments

This work was supported in part by Grant-in-Aid for Scientific Research (C)(20540125).

References

[1] M. Ajtai, R. Kumar, and D. Sivakumar. "A Sieve Algorithm for the Shortest Lattice Vector Problem." *STOC 2001*, pp. 266–275. (2001)

[2] H. Cohen. "A Course in Computational Algebraic Number Theory." GTM138, Springer. (1996)

[3] H. Cohen. "Advanced topics in computational number theory." GTM193, Springer. (1996)

[4] T. ElGamal. "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms." *IEEE Transactions on Information Theory*, Vol. IT-31, pp. 469–472. (1985)

[5] N. Gama and P. Q. Nguyen. "Predicting Lattice Reduction." *Eurocrypt '08*, LNCS 4965, pp. 31–51. (2008) Available at: `http://www.di.ens.fr/~pnguyen/pub_GaNg08.htm`.

[6] C. Gentry. "Fully Homomorphic Encryption Using Ideal Lattices." *STOC 2009*, pp. 169–178. (2009)

[7] C. Gentry. "A Fully Homomorphic Encryption Scheme." PhD thesis, Stanford University. (2009) Available at: `http://crypto.stanford.edu/craig`.

[8] C. Gentry and S. Halevi. "A Working Implementation of Fully Homomorphic Encryption." *Eurocrypt 2010 rump session*, `http://eurocrypt2010rump.cr.yp.to/9854ad3cab48983f7c2c5a2258e27717.pdf` in `http://eurocrypt2010rump.cr.yp.to`. (2010)

[9] O. Goldreich, S. Goldwasser, and S. Halevi. "Public-Key Cryptosystems from Lattice Reduction Problems." *Crypto '97*, LNCS1294, pp. 112–131. (1997)

[10] R. M. Gray. "Toeplitz and Circulant Matrices: A Review." *Foundation and Trends in Communications and Information Theory*, Vol. 2, No. 3, now Publishers Inc., USA. (2006)

[11] J. Hoffstein, J. Pipher, and J. Silverman. "NTRU: A Ring Based Public Key Cryptosystem." *ANTS III*, LNCS 1423, pp. 267–288. (1998)

[12] A. Y. Kitaev, A. H. Shen, and M. N. Vyalyi. "Classical and Quantum Computation." *Graduate Studies in Mathematics*, Vol. 47, AMS. (2002)

[13] A. K. Lenstra, H. W. Jr. Lenstra, and L. Lov'asz. "Factoring Polynomials with Rational Coefficients." *Mathematische Annalen*, 261, pp.513–534. (1982)

[14] D. Micciancio. "Improving Lattice-based Cryptosystems Using the Hermite Normal Form." *CALC '01*, LNCS 2146, pp. 126–145 (2001)

[15] N. Ogura, G. Yamamoto, T. Kobayashi, and S. Uchiyama. "An Improvement of Key Generation Algorithm for Gentry's Homomorphic Encryption Scheme." *IWSEC 2010*, LNCS 6434, pp. 70–83. (2010)

[16] T. Okamoto and S. Uchiyama. "A New Public-Key Cryptosystem as Secure as Factoring." *Eurocrypt '98*, LNCS 1403, pp. 308–318. (1998)

[17] P. Paillier. "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes." *Eurocrypt '99*, LNCS 1592, pp. 223–238. (1999)

[18] R. L. Rivest, A. Shamir, and L. Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." *Communications of ACM*, Vol.21, No. 2, pp. 120–126. (1978)

[19] C. P. Schnorr. "A Hierarchy of Polynomial Time Lattice Basis Reduction Algorithms." *Theoretical Computer Science*, 53(2-3), pp. 201–224. (1987)

[20] N. P. Smart and F. Vercauteren. "Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes." *PKC2010*, LNCS 6056, pp. 420–443. (2010) Available at: `http://eprint.iacr.org/2009/571`. (2009)

[21] D. Stehl'e and R. Steinfeld. "Faster Fully Homomorphic Encryption." *Cryptology ePrint archive*, `http://eprint.iacr.org/2010/299`. (2010)

[22] "Turing Machines." `http://www.math.ku.dk/~wester/turing.html`.

[23] Magma, `http://magma.maths.usyd.edu.au/magma/`

Naoki Ogura and Shigenori Uchiyama
Tokyo Metropolitan University, Tokyo 192-0397, Japan
E-mail: ogura-naoki(at)ed.tmu.ac.jp

Go Yamamoto and Tetsutaro Kobayashi
NTT Information Sharing Platform Laboratories, Tokyo 180-8585, Japan
E-mail: Not available