L-018

Lattice Cryptosystem with Polynomial Ring Secure Against the Han's Attack

藤堂 洋介 †　森井 昌克 †
Yosuke Todo　Masakatu Morii

1 Introduction

In 1997, Goldreich, Goldwasser and Halevil proposed a public key cryptosystem using the closest vector problem (CVP) [1]. We call this cryptosystem the GGH cryptosystem. It is a notable cryptosystem based on the complexity of lattices. However, its large key size is a bottleneck for practical use [2]. Then, several cryptosystems have been proposed to reduce the key size of the GGH cryptosystem. Micciancio introduced the Hermite normal form for the public key, and proposed a new encryption method in 2001 [3]. Paeng, Jung and Ha proposed a cryptosystem, we call it the PJH cryptosystem, by introducing a representation of a polynomial ring in 2003 [4]. Although the GGH cryptosystem requires the secret and public keys with $O(n^2)$ to encrypt a message with $O(n)$, the key sizes are only $O(n)$ in the PJH cryptosystem. Furthermore, the processing speed of the PJH cryptosystem is quicker than that of the GGH cryptosystem. However, Han et al. proposed a key recovery attack against the PJH cryptosystem using a special structure of a transformation matrix in the PJH cryptosystem [5]. According to the Han's attack, they succeeded to recover secret keys with $n = 1001$ on a single PC. As a result, the Han's attack ruins the practicality of the PJH cryptosystem. Hanawa, Kunihiro and Ohta improved the PJH cryptosystem by changing the generation of the transformation matrix without compromising the key size with $O(n)$ [6]. We call this cryptosystem the HKO cryptosystem. However, since the Euclidean norm of the public key is very large, the total amount of memory required for storing the key is too large. Furthermore, it is difficult to decrypt because of the large Euclidean norm of the ciphertext.

In this paper, we propose a diffusion matrix to be operated for the transformation matrix in the PJH cryptosystem. The proposed diffusion matrix excludes the special structure of the transformation matrix, and hence, the Han's attack is not applicable in the proposed cryptosystem. Moreover, the Euclidean norms of the public key and the ciphertext are about as large as that of the PJH cryptosystem. The advantage of the proposed cryptosystem is discussed under the consideration of possible lattice attacks. Then, our proposed cryptosystem is useful in a practical environment.

2 Lattice

In this paper, we only care about integral lattices of full rank. Let $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)^T$ be a non-singular $n \times n$ integral matrix. The lattice $\mathcal{L}(\boldsymbol{B})$ spanned by $\boldsymbol{B}$ is defined as follows:

$$\mathcal{L}(\boldsymbol{B}) = \mathcal{L}((\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n)^T) = \left\{ \sum_{i=1}^{n} x_i \boldsymbol{b}_i^T \ : \ x_i \in \mathbb{Z} \right\}.$$

The non-singular matrix $\boldsymbol{B}$ is called a lattice basis, and the determinant of $\mathcal{L}(\boldsymbol{B})$ is invariable. Namely, if a lattice basis $\boldsymbol{B}$ has the same determinant as the other basis $\boldsymbol{B}'$. Their lattices spanned by $\boldsymbol{B}$ and $\boldsymbol{B}'$ are the coincident with each other. Then there exists $\boldsymbol{T} \in \mathbb{Z}^{n \times n}$ such that $|\det(\boldsymbol{T})| = 1$ and $\boldsymbol{TB} = \boldsymbol{B}'$.

The CVP is a hard computational problem shown to be NP-hard. Given a lattice basis $\boldsymbol{B} \in \mathbb{Z}^{n \times n}$ and a target vector $\boldsymbol{t} \in \mathbb{Z}^n$, the CVP asks to find the lattice point closest to $\boldsymbol{t}$. The SVP is a hard computational problem closely related to the CVP. Given a lattice basis $\boldsymbol{B} \in \mathbb{Z}^{n \times n}$, the SVP asks to find the shortest non-zero lattice vector. In Euclidean norm, it is shown to be NP-hard for randomized reductions. However, a relatively short vector is introduced in polynomial time by using the lattice reduction algorithm, for example the LLL algorithm [7]. In most cases, one cannot prove that the lattice vector $\boldsymbol{v}$ is an exact shortest vector. Thus, one guesses the Euclidean norm of the shortest vector by using Gaussian heuristic [5] as follows:

$$\sigma(\mathcal{L}) \approx \sqrt{\frac{n}{2\pi e}} \det(\boldsymbol{B})^{\frac{1}{n}}.$$

If the Euclidean norm of the given vector is less than $\sigma(\mathcal{L})$, one may expect that this vector is the shortest vector. In practical, the larger $\sigma(\mathcal{L}) / \| \boldsymbol{v} \|$ is, the easier one can find $\boldsymbol{v}$ by using the lattice reduction algorithm, where $\| \boldsymbol{v} \|$ is the Euclidean norm of a vector $\boldsymbol{v}$. We call this ratio $\sigma(\mathcal{L}) / \| \boldsymbol{v} \|$ an "*expected gap*" of the lattice $\mathcal{L}$. Especially, if the expected gap is smaller than $1/\sqrt{2\pi e}$, the vector $\boldsymbol{v}$ is not the shortest vector by the Minkowski's theorem.

The lattice reduction algorithm is useful to solve the CVP. Given a lattice basis $\boldsymbol{B} \in \mathbb{Z}^{n \times n}$ and a target vector $\boldsymbol{t} \in \mathbb{Z}^n$, one generates a new lattice basis $\boldsymbol{B}' \in \mathbb{Z}^{n+1 \times n+1}$ as follows:

$$\boldsymbol{B}' = \begin{pmatrix} \boldsymbol{B} & \boldsymbol{0}_n \\ \boldsymbol{t} & 1 \end{pmatrix},$$

where $\boldsymbol{0}_n$ is a column vector of dimension n whose entries are all 0. Then, a vector $(-\boldsymbol{x}, 1)\boldsymbol{B}' = (\boldsymbol{t} - \boldsymbol{xB}, 1)$ is contained in $\mathcal{L}(\boldsymbol{B}')$. The CVP asks to find the lattice vector $\boldsymbol{xB}$ closest to the target $\boldsymbol{t}$. Then, if $\boldsymbol{t} - \boldsymbol{xB}$ is the shortest non-zero lattice vector in lattice $\mathcal{L}(\boldsymbol{B}')$, one can regard the CVP as the SVP and solve it by using the lattice reduction algorithm.

3 GGH series

GGH series are the variants of public key cryptosystems proposed in [1] based on lattice problems; for example the SVP or the CVP. In this section, we first describe the original GGH cryptosystem, and its variants the PJH cryptosystem and the HKO cryptosystem. Next, we describe their Euclidean norms of the public key and the ciphertext. Finally, we describe that the Euclidean norm of the ciphertext makes it difficult to decrypt in the HKO cryptosystem.

3.1 The GGH cryptosystem

The GGH cryptosystem uses a non-singular matrix $\boldsymbol{R} \in \mathbb{Z}^{n \times n}$ as a secret key. First, $\boldsymbol{R}$ is generated as follows:

$$\boldsymbol{R} = k\boldsymbol{I} + \boldsymbol{R}',$$

where $k\boldsymbol{I}$($k = \sqrt{n}l$) is an orthogonal matrix and a matrix $\boldsymbol{R}'$ is uniformly distributed in $\{-l, \ldots, l\}^{n \times n}$. Next, a public key $\boldsymbol{B}$ is generated as follows:

$$\boldsymbol{B} = \boldsymbol{TR},$$

where $\boldsymbol{T}$ is a transformation matrix and $|\det(\boldsymbol{T})| = 1$. In this case, their lattices spanned by the secret key $\boldsymbol{R}$ and the public key $\boldsymbol{B}$ are the coincident with each other. Micciancio proposed another method of generating the public key from the Hermite normal form

† 神戸大学大学院工学研究科，Graduate School of Engineering, Kobe University

of the secret key. Its public key size is smaller than that of the GGH cryptosystem.

One chooses a lattice vector from $\boldsymbol{mB}$, where the vector $\boldsymbol{m}$ is chosen at random from $\mathbb{Z}^n$. The ciphertext $\boldsymbol{c} \in \mathbb{Z}^n$ is calculated by

$$\boldsymbol{c} = \boldsymbol{mB} + \boldsymbol{e},$$

where $\boldsymbol{e}$ is an error vector and its coefficients are contained in $\{-\sigma, \sigma\}$. In GGH series, there are two methods to encode a message. In a first method, a message is encoded in $\boldsymbol{m}$ which is used for choice of a lattice vector. On the other hand, another method encodes a message in $\boldsymbol{e}$ which is used for an error vector.

The ciphertext is decrypted by using the Babai's rounding algorithm [8] as follows:

$$\begin{aligned} \lfloor \boldsymbol{cR}^{-1} \rceil \boldsymbol{T}^{-1} &= \lfloor (\boldsymbol{mTR} + \boldsymbol{e})\boldsymbol{R}^{-1} \rceil \boldsymbol{T}^{-1} \\ &= \lfloor \boldsymbol{mT} + \boldsymbol{eR}^{-1} \rceil \boldsymbol{T}^{-1} \\ &= \boldsymbol{m} + \lfloor \boldsymbol{eR}^{-1} \rceil \boldsymbol{T}^{-1}. \end{aligned} \tag{1}$$

From Eq. (1), it is noticed that the decryption works well only when $\lfloor \boldsymbol{eR}^{-1} \rceil = 0$. This holds with high probability since $\boldsymbol{R}^{-1}$ consists of very small values. An alternative to the Babai's rounding algorithm is the nearest plane algorithm [8]. This algorithm solves the CVP by using the Gram-Schmidt basis of the secret key and derives the lattice vector $\boldsymbol{mB}$.

3.2 The PJH cryptosystem

The PJH cryptosystem is a special case of the GGH cryptosystem. It introduces a representation of a polynomial ring, and its key size is reduced from that of the GGH cryptosystem. They use a polynomial ring $\mathcal{R} = \mathbb{Z}[x]/(x^N - 1)$. Note that the multiplication $\boldsymbol{f} \cdot \boldsymbol{g} \in \mathcal{R}$ of $\boldsymbol{f}$ and $\boldsymbol{g}$ is computed by the convolution product of them, that is,

$$\boldsymbol{h} = \boldsymbol{f}\, \Phi(\boldsymbol{g}),$$

where $\Phi(\boldsymbol{g})$ is an $(N \times N)$ circulant matrix of vector $\boldsymbol{g} \in \mathbb{Z}^N$. So arithmetic operations of the PJH cryptosystem are defined under this polynomial ring.

The secret key of the PJH cryptosystem is $\{\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{h}_1, \boldsymbol{h}_2 \in \mathcal{R}\}$, which have the following properties:

- $\boldsymbol{f}_1(x) = \alpha_{N-1}x^{N-1} + \cdots + \alpha_0$ and $\boldsymbol{f}_2(x) = \beta_{N-1}x^{N-1} + \cdots + \beta_0$, where $|\alpha_{i0}|, |\beta_{j0}| \approx \sqrt{2N}$ for some i_0, j_0 and the other coefficients are contained in $\{-1, 0, 1\}$.
- The coefficients of $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ are contained in $\{-1, 0, 1\}$.

Then, the secret basis $\boldsymbol{R}$ is given as follows:

$$\boldsymbol{R} = \begin{pmatrix} \Phi(\boldsymbol{f}_1) & \Phi(\boldsymbol{h}_2) \\ \Phi(\boldsymbol{h}_1) & \Phi(\boldsymbol{f}_2) \end{pmatrix}.$$

The public basis $\boldsymbol{B}$ is the product of the transformation matrix $\boldsymbol{T}$ and the secret basis $\boldsymbol{R}$. To represent the public basis $\boldsymbol{B}$ by a circulant matrix, the transformation matrix $\boldsymbol{T}$ is represented by a circulant matrix as follows:

$$\boldsymbol{T} = \begin{pmatrix} \Phi(\boldsymbol{g}) & \Phi(\boldsymbol{Q}) \\ p\boldsymbol{I} & \Phi(\boldsymbol{g}_p) \end{pmatrix}.$$

In order to generate the transformation matrix, one first chooses $\boldsymbol{g} \in \mathcal{R}$ such that the coefficients of $\boldsymbol{g}$ are contained in $(-p/2, p/2]$, where p is a random positive integer. Then, $\boldsymbol{g}$ can be considered as an element of a ring $\mathbb{Z}_p[x]/(x^N - 1)$, and one takes $\boldsymbol{g}$ which is invertible in this ring. Next, one calculates $\boldsymbol{g}_p$ such that $\boldsymbol{g} \cdot \boldsymbol{g}_p = 1 \in \mathbb{Z}_p[x]/(x^N - 1)$. Then, there exists $\boldsymbol{Q}$ such that $\boldsymbol{g} \cdot \boldsymbol{g}_p - p\boldsymbol{Q} = 1 \in \mathcal{R}$. Finally, from $\det(\boldsymbol{T}) = \det(\Phi(\boldsymbol{g} \cdot \boldsymbol{g}_p - p\boldsymbol{Q})) = \det(\boldsymbol{I}) = 1$, the determinant of this transformation matrix is 1. Therefore one generates the public basis $\boldsymbol{B}$ as follows:

$$\boldsymbol{B} = \boldsymbol{TR} = \begin{pmatrix} \Phi(\boldsymbol{P}_1) & \Phi(\boldsymbol{P}_3) \\ \Phi(\boldsymbol{P}_2) & \Phi(\boldsymbol{P}_4) \end{pmatrix},$$

where $\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_3, \boldsymbol{P}_4 \in \mathcal{R}$ are represented by

$$\boldsymbol{P}_1 = \boldsymbol{f}_1 \cdot \boldsymbol{g} + \boldsymbol{h}_1 \cdot \boldsymbol{Q} \tag{2}$$

$$\boldsymbol{P}_2 = p\boldsymbol{f}_1 + \boldsymbol{h}_1 \cdot \boldsymbol{g}_p \tag{3}$$

$$\boldsymbol{P}_3 = \boldsymbol{h}_2 \cdot \boldsymbol{g} + \boldsymbol{f}_2 \cdot \boldsymbol{Q} \tag{4}$$

$$\boldsymbol{P}_4 = p\boldsymbol{h}_2 + \boldsymbol{f}_2 \cdot \boldsymbol{g}_p. \tag{5}$$

The secret key of the PJH cryptosystem is the 4 polynomials $\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{h}_1$ and $\boldsymbol{h}_2$, and the public key is the 4 polynomials $\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_3$ and $\boldsymbol{P}_4$. Moreover, the 3 polynomials $\boldsymbol{g}, \boldsymbol{g}_p$ and $\boldsymbol{Q}$ are secret parameters, but even if the positive integer p is not a secret parameter, it does not seem to be a critical parameter for the security.

Let $\boldsymbol{m} = (\boldsymbol{m}_1, \boldsymbol{m}_2) \in \mathcal{R}^2$ be a message. Then, the ciphertext $\boldsymbol{c} = (\boldsymbol{c}_1, \boldsymbol{c}_2)$ is calculated by

$$\begin{aligned} (\boldsymbol{c}_1, \boldsymbol{c}_2) &= (\boldsymbol{m}_1, \boldsymbol{m}_2) \begin{pmatrix} \Phi(\boldsymbol{P}_1) & \Phi(\boldsymbol{P}_3) \\ \Phi(\boldsymbol{P}_2) & \Phi(\boldsymbol{P}_4) \end{pmatrix} + (\boldsymbol{e}_1, \boldsymbol{e}_2) \\ &= (\boldsymbol{m}_1 \cdot \boldsymbol{P}_1 + \boldsymbol{m}_2 \cdot \boldsymbol{P}_2 + \boldsymbol{e}_1,\ \boldsymbol{m}_1 \cdot \boldsymbol{P}_3 + \boldsymbol{m}_2 \cdot \boldsymbol{P}_4 + \boldsymbol{e}_2), \end{aligned}$$

where $\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2)$ is an error vector and its coefficients are contained in $\{-1/2, 1/2\}$. In the PJH cryptosystem, the decryption works by the same reason as the GGH cryptosystem. It requires the secret and public keys with only $O(n)$ to encrypt a message with $O(n)$. Furthermore, its processing speed is quicker than that of the GGH cryptosystem.

Although the PJH cryptosystem improved the practicality of the GGH cryptosystem, an efficient key recovery attack has been proposed by Han et al [5]. In the attack, they introduced the following equation under a ring $\mathcal{R}$:

$$\begin{aligned} \boldsymbol{g} \cdot \boldsymbol{P}_2 &= p\boldsymbol{f}_1 \cdot \boldsymbol{g} + \boldsymbol{h}_1 \cdot \boldsymbol{g}_p \cdot \boldsymbol{g} \\ &= p\boldsymbol{f}_1 \cdot \boldsymbol{g} + \boldsymbol{h}_1 \cdot (1 + p\boldsymbol{Q}) \\ &= p(\boldsymbol{f}_1 \cdot \boldsymbol{g} + \boldsymbol{h}_1 \cdot \boldsymbol{Q}) + \boldsymbol{h}_1 \\ &= p\boldsymbol{P}_1 + \boldsymbol{h}_1. \end{aligned}$$

Then, they generate the lattice spanned by the following basis $\boldsymbol{B}'$:

$$\boldsymbol{B}' = \begin{pmatrix} \Phi(\boldsymbol{P}_2) & \boldsymbol{0}_n \\ \boldsymbol{P}_1 & 1 \end{pmatrix},$$

where this basis is $(N + 1) \times (N + 1)$ matrix, and a short vector

$$\boldsymbol{v} = (\boldsymbol{g}, -p)\boldsymbol{B}' = (\boldsymbol{h}_1, -p)$$

is contained in $\mathcal{L}(\boldsymbol{B}')$. If they get this short vector $\boldsymbol{v}$ by executing the lattice reduction algorithm against $\boldsymbol{B}'$, they can recover the transformation matrix and the secret key.

3.3 The HKO cryptosystem

Hanawa, Kunihiro and Ohta proposed a new method of generating the transformation matrix. The Han's attack uses a special structure of the transformation matrix in the PJH cryptosystem, in particular the block matrix $p\boldsymbol{I}$. Instead of the transformation matrix of the PJH cryptosystem, they generate the transformation matrix from 4 polynomials $\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3$ and $\boldsymbol{g}_4$ as follows:

$$\boldsymbol{T} = \begin{pmatrix} \Phi(\boldsymbol{g}_1) & \Phi(\boldsymbol{g}_3) \\ \Phi(\boldsymbol{g}_2) & \Phi(\boldsymbol{g}_4) \end{pmatrix}.$$

In order to generate the transformation matrix, one first chooses 2 polynomials $\boldsymbol{a}$ and $\boldsymbol{b}$ such that the coefficients of $\boldsymbol{a}$ and $\boldsymbol{b}$ are contained in $\{-1, 0, 1\}$. Next, one calculates the determinants R_a and R_b for $\Phi(\boldsymbol{a})$ and $\Phi(\boldsymbol{b})$, respectively. Then, there exist $\boldsymbol{s}, \boldsymbol{t}, \boldsymbol{s}', \boldsymbol{t}' \in \mathbb{Z}[x]$ such that $\boldsymbol{a} \cdot \boldsymbol{s} + (x^N - 1) \cdot \boldsymbol{t} = R_a$ and

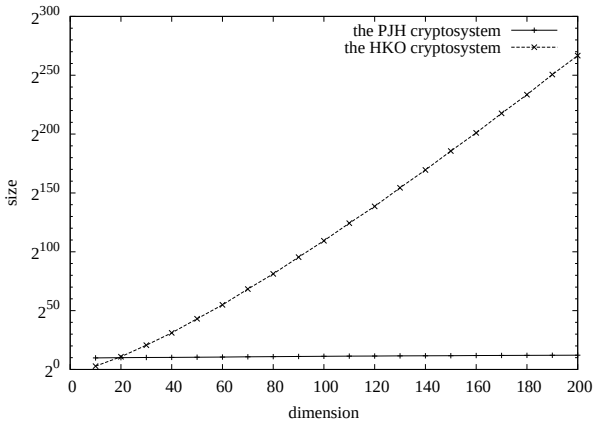

Fig.1 Experimental results about the Euclidean norm of $\boldsymbol{mT}$.

$\boldsymbol{b} \cdot \boldsymbol{s}' + (x^N - 1) \cdot \boldsymbol{t}' = R_b$. If $\gcd(R_a, R_b) = 1$, the Extended Euclidean Algorithm returns $u, v \in \mathbb{Z}$ such that $R_a u + R_b v = 1$. Finally, one introduces $(u\boldsymbol{a}) \cdot \boldsymbol{s} + (v\boldsymbol{b}) \cdot \boldsymbol{s}' = 1 \in \mathcal{R}$, then let $\boldsymbol{g}_1 = u\boldsymbol{a}, \boldsymbol{g}_2 = -v\boldsymbol{b}, \boldsymbol{g}_3 = \boldsymbol{s}'$ and $\boldsymbol{g}_4 = \boldsymbol{s}$, respectively.

Although the determinant of this transformation matrix is 1, the Euclidean norms of 4 polynomials are much larger than that of the PJH cryptosystem because of the following reason. Each coefficient of 4 polynomials is bounded by the determinants R_a and R_b. The size of R_a and R_b are $O(\| \boldsymbol{a} \|^N)$ and $O(\| \boldsymbol{b} \|^N)$, respectively. Then, the public key size of the HKO cryptosystem is much larger than that of the PJH cryptosystem. For example, in the PJH cryptosystem with $N = 100$ (this dimension is 200), the public key size is about $0.85KB$, on the other hand, in the HKO cryptosystem with $N = 100$ (this dimension is 200), that is $12KB$. Moreover, in the PJH and HKO cryptosystem, the ciphertext size almost becomes equal with the public key size. The ciphertext size of the HKO cryptosystem is serious problem, because it exponentially increased with the dimension.

4 Difficulty of decryption

In Sect. 3.1, there are two methods to encode a message; one encodes it in $\boldsymbol{m}$ and the other in $\boldsymbol{e}$. In both methods, all operations are mathematically defined under a ring $\mathcal{R}$. However, in a real environment, it is difficult for a computer to simulate operations because of the limitation of its precision. A rounding error at an operation may critically changes the results of decryption. For example, if the decryption is executed by using the Babai's rounding algorithm, the inverse matrix of the secret basis $\boldsymbol{R}$ is calculated and may have a tiny error $\boldsymbol{R}'$ as follows:

$$\boldsymbol{R} \times \boldsymbol{R}^{-1} = \boldsymbol{I} + \boldsymbol{R}'.$$

If the Euclidean norm of the ciphertext is large and the inverse matrix has a tiny error $\boldsymbol{R}'$, the decryption in Eq. (1) must be replaced as follows:

$$\begin{aligned} \lfloor \boldsymbol{c}\boldsymbol{R}^{-1} \rceil \boldsymbol{T}^{-1} &= \lfloor (\boldsymbol{m}\boldsymbol{T}\boldsymbol{R} + \boldsymbol{e})\boldsymbol{R}^{-1} \rceil \boldsymbol{T}^{-1} \\ &= \lfloor \boldsymbol{m}\boldsymbol{T} + \boldsymbol{m}\boldsymbol{T}\boldsymbol{R}' + \boldsymbol{e}\boldsymbol{R}^{-1} \rceil \boldsymbol{T}^{-1} \qquad (6) \\ &= \boldsymbol{m} + \lfloor \boldsymbol{m}\boldsymbol{T}\boldsymbol{R}' + \boldsymbol{e}\boldsymbol{R}^{-1} \rceil \boldsymbol{T}^{-1}. \end{aligned}$$

From Eq. (6), it is noticed that the decryption fails if the Euclidean norm of $\boldsymbol{mT}$ is larger than expected, where the Euclidean norm of $\boldsymbol{mT}$ is the maximum Euclidean norm of the column vectors of $\boldsymbol{mT}$.

Figure 1 shows experimental results about the Euclidean norm of $\boldsymbol{mT}$ in the PJH and the HKO cryptosystems with dimensions 10-200. In the HKO cryptosystem with $N = 100$ and 200 dimensions, the Euclidean norm of $\boldsymbol{mT}$ becomes almost 2^{270}. Therefore, the error of the inverse matrix must be smaller than 2^{-270}, because the decryption fails if $\lfloor \boldsymbol{m}\boldsymbol{T}\boldsymbol{R}' \rceil \neq 0$. However, the calculation of the inverse matrix with high accuracy requires many computer resources. Moreover, the required precision is exponentially increased with the dimension.

5 The proposed method

The HKO cryptosystem can immunize the Han's attack, but the Euclidean norm of the ciphertext is much larger than that of the PJH cryptosystem. In this section, we propose a diffusion matrix which excludes the special structure of the transformation matrix of the PJH cryptosystem. The proposed cryptosystem can immunize the Han's attack, and the Euclidean norms of the public key and the ciphertext are about as large as that of the PJH cryptosystem.

5.1 The proposed transformation marix

To immunize the Han's attack, a transformation matrix $\boldsymbol{T}$ is given as follows:

$$\boldsymbol{T} = \begin{pmatrix} \Phi(\boldsymbol{g}) & \Phi(\boldsymbol{Q}) \\ \Phi(p + \boldsymbol{A} \cdot \boldsymbol{g}) & \Phi(\boldsymbol{g}_p + \boldsymbol{A} \cdot \boldsymbol{Q}) \end{pmatrix}.$$

In order to generate the transformation matrix, the 3 polynomials $\boldsymbol{g}, \boldsymbol{g}_p$ and $\boldsymbol{Q}$ and a positive integer p is generated similar to that of the PJH cryptosystem. However, the coefficients of $\boldsymbol{g}$ are contained in $[-\ell, \ell]$ not $(-p/2, p/2]$ ($\ell \leq \lfloor p/2 \rfloor$), and the reason is discussed later. Next, a polynomial $\boldsymbol{A}$ is generated at random in a ring $\mathcal{R}$. Finally, $p + \boldsymbol{A} \cdot \boldsymbol{g} \in \mathcal{R}$ and $\boldsymbol{g}_p + \boldsymbol{A} \cdot \boldsymbol{Q} \in \mathcal{R}$ are generated to immunize the Han's attack, and $\boldsymbol{A}$ is secret parameter and is disused after the abovementioned operations.

The determinant of the proposed transformation matrix is 1 because

$$\begin{aligned} \begin{pmatrix} \Phi(\boldsymbol{g}) & \Phi(\boldsymbol{Q}) \\ \Phi(p + \boldsymbol{A} \cdot \boldsymbol{g}) & \Phi(\boldsymbol{g}_p + \boldsymbol{A} \cdot \boldsymbol{Q}) \end{pmatrix} &= \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \Phi(\boldsymbol{A}) & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} \Phi(\boldsymbol{g}) & \Phi(\boldsymbol{Q}) \\ p\boldsymbol{I} & \Phi(\boldsymbol{g}_p) \end{pmatrix} \\ &= \boldsymbol{A}' \boldsymbol{T}_{pjh}, \end{aligned}$$

where $\boldsymbol{A}'$ is a lower triangular matrix whose determinant is 1 and $\boldsymbol{T}_{pjh}$ is a transformation matrix of the PJH cryptosystem. We call this operation the diffusion of the transformation matrix and the $\boldsymbol{A}'$ is the diffusion matrix. A public basis $\boldsymbol{B}$ is calculated as follows:

$$\boldsymbol{B} = \boldsymbol{T}\boldsymbol{R} = \begin{pmatrix} \Phi(\boldsymbol{P}_1) & \Phi(\boldsymbol{P}_3) \\ \Phi(\boldsymbol{P}_2) & \Phi(\boldsymbol{P}_4) \end{pmatrix},$$

where $\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_3, \boldsymbol{P}_4 \in \mathcal{R}$ are expressed as

$$\begin{aligned} \boldsymbol{P}_1 &= \boldsymbol{P}_{1pjh} \\ \boldsymbol{P}_2 &= \boldsymbol{P}_{2pjh} + \boldsymbol{A} \cdot \boldsymbol{P}_{1pjh} \\ \boldsymbol{P}_3 &= \boldsymbol{P}_{3pjh} \\ \boldsymbol{P}_4 &= \boldsymbol{P}_{4pjh} + \boldsymbol{A} \cdot \boldsymbol{P}_{3pjh}, \end{aligned}$$

where $\{\boldsymbol{P}_{1pjh}, \boldsymbol{P}_{2pjh}, \boldsymbol{P}_{3pjh}, \boldsymbol{P}_{4pjh}\}$ is the public key of the PJH cryptosystem. It is noticed from the above public basis $\boldsymbol{B}$ that the Euclidean norm of the ciphertext is almost equal to that of the PJH cryptosystem.

The Han's attack decreases the dimension of the lattice problem by using the block matrix $p\boldsymbol{I}$. However, the proposed method covers the block matrix $p\boldsymbol{I}$ with $\boldsymbol{A} \cdot \boldsymbol{g}$. Then, the Han's attack is expressed as follows:

$$\begin{aligned} \boldsymbol{g} \cdot \boldsymbol{P}_2 &= \boldsymbol{g} \cdot (\boldsymbol{P}_{2pjh} + \boldsymbol{A} \cdot \boldsymbol{P}_{1pjh}) \\ &= p\boldsymbol{P}_{1pjh} + \boldsymbol{h}_1 + \boldsymbol{g} \cdot \boldsymbol{A} \cdot \boldsymbol{P}_{1pjh} \\ &= (p + \boldsymbol{g} \cdot \boldsymbol{A}) \cdot \boldsymbol{P}_1 + \boldsymbol{h}_1, \end{aligned}$$

Table 1　The experimental result of expected gap

dimension	target lattice	$\ell = p/2$		$\ell = p/\sqrt{N}$		$\ell = p/N$	
		expected gap	success/trial	expected gap	success/trial	expected gap	success/trial
100	$(\boldsymbol{P}_{12}, \boldsymbol{v}_2)$	2.020990949	9/10	0.427815336	0/10	0.073482035	0/10
	$(\boldsymbol{P}_{34}, \boldsymbol{v}_4)$	1.783193973	9/10	0.40692057	0/10	0.066239996	0/10
140	$(\boldsymbol{P}_{12}, \boldsymbol{v}_2)$	2.612034438	10/10	0.608524053	0/10	0.069415363	0/10
	$(\boldsymbol{P}_{34}, \boldsymbol{v}_4)$	2.172856764	10/10	0.54451621	0/10	0.061149658	0/10
180	$(\boldsymbol{P}_{12}, \boldsymbol{v}_2)$	3.183394326	10/10	0.671688273	0/10	0.070966117	0/10
	$(\boldsymbol{P}_{34}, \boldsymbol{v}_4)$	3.118077292	10/10	0.6032438	0/10	0.066959833	0/10
220	$(\boldsymbol{P}_{12}, \boldsymbol{v}_2)$	4.056582215	10/10	0.758704508	0/10	0.067518278	0/10
	$(\boldsymbol{P}_{34}, \boldsymbol{v}_4)$	3.890429631	10/10	0.717111322	0/10	0.064699121	0/10

where $p + \boldsymbol{g} \cdot \boldsymbol{A}$ is a vector under a ring $\mathcal{R}$ not a positive integer. Then, an adversary cannot decrease the dimension of the lattice problem by the Han's attack.

5.2 Lattice attacks against the proposed cryptosystem

The proposed cryptosystem can immunize the Han's attack. However if an adversary can recover the public key of the PJH cryptosystem from that of the proposed cryptosystem, one can execute the Han's attack after the recovery of the public key of the PJH cryptosystem. Then, we consider the method to recover the public key of the PJH cryptosystem from that of the proposed cryptosystem.

In order to recover the public key of the PJH cryptosystem, one must calculate the following equation,

$$\begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \Phi(-\boldsymbol{A}) & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} \Phi(\boldsymbol{P}_1) & \Phi(\boldsymbol{P}_3) \\ \Phi(\boldsymbol{P}_2) & \Phi(\boldsymbol{P}_4) \end{pmatrix} = \begin{pmatrix} \Phi(\boldsymbol{P}_{1pjh}) & \Phi(\boldsymbol{P}_{3pjh}) \\ \Phi(\boldsymbol{P}_{2pjh}) & \Phi(\boldsymbol{P}_{4pjh}) \end{pmatrix},$$

then,

$$-\boldsymbol{A}\Phi(\boldsymbol{P}_1) + \boldsymbol{P}_2 = \boldsymbol{P}_{2pjh} \quad (7)$$

$$-\boldsymbol{A}\Phi(\boldsymbol{P}_3) + \boldsymbol{P}_4 = \boldsymbol{P}_{4pjh}, \quad (8)$$

where an adversary can know only $\boldsymbol{P}_1, \boldsymbol{P}_2, \boldsymbol{P}_3$ and $\boldsymbol{P}_4$. If $\boldsymbol{A}$ is derived from Eq. (7) and Eq. (8), the public key of the PJH cryptosystem is recovered. However, it is difficult to solve these equations. Assuming that these problems are the SVP of the following lattice bases,

$$\boldsymbol{P}_{12} = \begin{pmatrix} \Phi(\boldsymbol{P}_1) & \boldsymbol{0}_n \\ \boldsymbol{P}_2 & 1 \end{pmatrix} \text{ or } \boldsymbol{P}_{34} = \begin{pmatrix} \Phi(\boldsymbol{P}_3) & \boldsymbol{0}_n \\ \boldsymbol{P}_4 & 1 \end{pmatrix},$$

where these bases are $(N+1) \times (N+1)$ matrix, and lattice vectors $\boldsymbol{v}_2 = (\boldsymbol{P}_{2pjh}, 1)$ and $\boldsymbol{v}_4 = (\boldsymbol{P}_{4pjh}, 1)$ are contained in $\mathcal{L}(P_{12})$ and $\mathcal{L}(P_{34})$, respectively. If the lattice reduction algorithm derives $\boldsymbol{v}_2$ or $\boldsymbol{v}_4$, the proposed cryptosystem is not secure. To confirm it, we implemented this lattice attack using the NTL library [9], and evaluated the resistance to this lattice attack from the perspective of the expected gap. We used the BKZ_FP function whose block size is 15 in the NTL library.

We experimented by 3 parameters, $\ell = p/2$, $\ell = p/\sqrt{N}$ and $\ell = p/N$. Generally, the smaller coefficients of $\boldsymbol{g}$ are, the smaller that of $\boldsymbol{P}_1$ and $\boldsymbol{P}_3$ becomes from Eq. (2) and Eq. (4), but that of $\boldsymbol{P}_{2pjh}$ and $\boldsymbol{P}_{4pjh}$ do not become small from Eq. (3) and Eq. (5). Then, the smaller the ℓ is, the more difficult these lattice problems become because the determinants of these lattices grow smaller. In the experiment, p takes a random 10-bit integer. Table 1 shows these experimental results.

We can recover all the public keys of the PJH cryptosystem from that of the proposed cryptosystem when $\ell = p/2$. However we cannot recover all those if $\ell \leq p/\sqrt{N}$. Moreover, from the expected gaps when $\ell = p/N$, the lattice vectors $\boldsymbol{v}_2$ and $\boldsymbol{v}_4$ are not the shortest non-zero lattice vectors in lattices $\mathcal{L}(\boldsymbol{P}_{12})$ and $\mathcal{L}(\boldsymbol{P}_{34})$ from the Minkowski's theorem. In the proposed cryptosystem, coefficients of $\boldsymbol{g}$ takes are smaller than that of the PJH cryptosystem on average. This information may be helpful to attack. However, if p takes a large positive integer, this problem can be disregarded.

6 Conclusion

In this paper, we improved the PJH cryptosystem by diffusing the transformation matrix, and it can immunize the Han's attack. Moreover, the Euclidean norm of the ciphertext is about as large as that of the PJH cryptosystem. We showed that if the coefficients of $\boldsymbol{g}$ is selected appropriately, it was difficult to recover the public key of the PJH cryptosystem from that of the proposed cryptosystem. On the other hand, we experimentally derive the coefficients of $\boldsymbol{g}$ in this paper. The theoretical analysis about the appropriate coefficients of $\boldsymbol{g}$ is left for our future work.

The security of the proposed cryptosystem is not proved. However, the security of the proposed cryptosystem may be an equal to that of the PJH cryptosystem secure against the Han's attack.

Reference

[1] O. Goldreich, S. Goldwasser, and S. Halevi, "Public-Key Cryptosystems from Lattice Reduction Problems," CRYPTO, LNCS, vol.1294, pp.112–131, 1997.

[2] P.Q. Nguyen, "Cryptanalysis of the Goldreich-Goldwasser-Halevi Cryptosystem from Crypto '97," CRYPTO, LNCS, vol.1666, pp.288–304, 1999.

[3] D. Micciancio, "Improving Lattice based cryptosystems using the Hermite Normal Form," CaLC 2001, LNCS, vol.2146, pp.126–145, 2001.

[4] S.H. Paeng, B.E. Jung, and K.C. Ha, "A Lattice Based Public Key Cryptosystem Using Polynomial Representations," Public Key Cryptography, LNCS, vol.2567, pp.292–308, 2003.

[5] D. Han, M.H. Kim, and Y. Yeom, "Cryptanalysis of the Paeng-Jung-Ha cryptosystem from PKC 2003," Public Key Cryptography, LNCS, vol.4450, pp.107–117, 2007.

[6] T. Hanawa, N. Kunihiro, and K. Ohta, "Improvement of a Lattice Based Cryptosystem Using Polynomial Ring." SCIS 2009, 2009.

[7] A.K. Lenstra, H.W. Lenstra, and L. Lovasz, "Factoring polynomials with rational coefficients," Mathematische Annalen, vol.261, pp.515–534, 1982.

[8] L. Babai, "On Lovasz' lattice reduction and the nearest lattice point problem," Combinatorica, vol.6, pp.1–13, 1986.

[9] V. Shoup, "NTL: A Library for doing Number Theory." `http://www.shoup.net/ntl/`.