A Public-key Encryption Scheme Based on Non-linear Indeterminate Equations (Giophantus)

Koichiro Akiyama[1], Yasuhiro Goto[2], Shinya Okumura[3⋆], Tsuyoshi Takagi[4], Koji Nuida[5], Goichiro Hanaoka[5], Hideo Shimizu[1], and Yasuhiko Ikematsu[4]

[1] Corporate Research& Development Center, Toshiba Corporation
`{koichiro.akiyama,hideo.shimizu}@toshiba.co.jp`
[2] Department of Mathematics, Hokkaido University of Education at Hakodate
`goto.yasuhiro@h.hokkyodai.ac.jp`
[3] Department of Information and Communications Technology, Osaka University
`okumura@cy2sec.comm.eng.osaka-u.ac.jp`
[4] Institute of Mathematics for Industry, Kyushu University
`{takagi,y-ikematsu}@imi.kyushu-u.ac.jp`
[5] National Institute of Advanced Industrial Science and Technology
`{k.nuida,hanaoka-goichiro}@aist.go.jp`

Abstract. In this paper, we propose a post-quantum public-key encryption scheme whose security depends on a problem arising from a multivariate non-linear indeterminate equation. The security of lattice cryptosystems, which are considered to be the most promising candidate for a post-quantum cryptosystem, is based on the shortest vector problem or the closest vector problem in the discrete linear solution spaces of simultaneous equations. However, several improved attacks for the underlying problems have recently been developed by using approximation methods, which result in requiring longer key sizes. As a scheme to avoid such attacks, we propose a public-key encryption scheme based on the "smallest" solution problem in the *non-linear* solution spaces of multivariate *indeterminate equations* that was developed from the algebraic surface cryptosystem. Since no efficient algorithm to find such a smallest solution is currently known, we introduce a new computational assumption under which proposed scheme is proven to be secure in the sense of IND-CPA. Then, we perform computational experiments based on known attack methods and evaluate that the key size of our scheme under the linear condition. This paper is a revised version of [4].

Keywords: Public-key Cryptosystem, Post-Quantum Cryptosystem, Indeterminate Equation, Smallest Solution Problem

⋆ Research conducted while at Institute of Mathematics for Industry, Kyushu University

1 Introduction

In 1994, Shor proposed quantum algorithms that can solve the factorization problem and the discrete logarithm problem in polynomial time [47]. This implies that elliptic curve cryptosystems and the RSA cryptosystem will no longer be secure once a quantum computer is built. Due to this, the importance of "Post-quantum cryptosystems" (PQCs) that will still be secure after the development of quantum computers has been recognized. With the recent active studies to develop quantum computers, NIST announced that the process of PQC standardization will begin in the end of 2017 [40]. Possible candidates for a PQC include lattice-based encryptions, code-based encryptions, and multivariate encryptions.

First lattice-based encryption was proposed in 1997 by Ajtai and Dwork [1]. Its security depends on the unique shortest vector problem in lattices. Goldreich et al. proposed the GGH cryptosystem, whose security is based on the closest vector problem for an integer lattice [24]. However, according to Nguyen and Stern, these schemes are not practical since they require large size parameters for security reasons [36, 35]. Hoffstein et al. proposed the NTRU cryptosystem, whose security depends on the shortest vector problem for polynomial ring lattices [25]. In 2009, Regev proposed an LWE cryptosystem, whose security depends on the "learning with error" (LWE) problem [45]. Currently, NTRU, LWE, and their variants are relatively efficient among lattice-based encryption schemes.

However, there are several efficient approximation algorithms for finding the (nearly) shortest/closest vectors, such as the LLL [30], BKZ [46], and BKZ2.0 [15] algorithms. Recently, several improved attacks for these underlying problems using these methods, such as lattice decoding attacks [10] and subfield lattice attacks [28] have been developed. In order to avoid these attacks, the public-key sizes of lattice-based cryptosystems must be enlarged. Encryption schemes with large key sizes require a large amount of memory in applications.

Code-based encryption was first proposed in 1978 by McEliece [33]. Its security depends on the decoding problem for random linear codes, for which only exponential algorithms are known. However, it requires a large public-key size, of more than 1M bits. The multivariate public-key cryptosystem (MPKC) was first introduced in 1989 by Matsumoto and Imai [26] and was improved by Patarin [43]. Its security depends on the problem of solving non-linear equations (called multivariate equations) over finite fields. While the problem is NP-hard in general, almost all proposed schemes have been broken due to the special structure of the equations that are used as public keys. Several schemes with resistance against known attacks on MPKC have been proposed, but they still have large public keys [44, 49, 53].

These candidates require large public-key sizes of more than 24K bits (under 128-bit security) to avoid improved attacks that take advantage of the special structure of the schemes. Even though many PQC candidates have been proposed, none of them are efficient enough for practical use. This might be due to their large public-key sizes and the large amount of memory that is therefore required in applications. In an effort to find a more practical PQC, Akiyama et al.

proposed the algebraic surface cryptosystem (ASC) [3], whose security depends on the section-finding problem (the problem of solving some kind of indeterminate equation). Although they claimed that their proposed scheme necessitates much shorter public keys than the other candidates for PQC, the scheme was broken by Faugére et al. [19]. In this paper, we intend to improve ASC by modifying the underlying problem to make the scheme secure while keeping the public-key size small relative to that of other PQC candidates.

Our Contribution. This paper proposes a post-quantum public-key encryption scheme whose security is based on the smallest solution problem for non-linear solution spaces of indeterminate equations, to which attack algorithms based on approximation (e.g., LLL and BKZ) cannot be applied. Our scheme was developed from ASC, which is designed such that its security depends on the intractability of solving some non-linear indeterminate equation [3]. ASC was broken by the ideal decomposition attack proposed in PKC 2010 [19]. We revise the scheme to be secure against this attack by adding a noise term to the cipher polynomial. Our scheme is provably secure in the sense of IND-CPA under the intermediate equation of LWE (IE-LWE) assumption, which is a new computational assumption coming from analogy to the LWE assumption. An IND-CCA2 secure scheme is obtained by using a well-known conversion technique [20].

We refer to the public key encryption scheme as the **Giophantus™ encryption scheme**, which comes from the Diophantine equations used as the general term for the indeterminate equations in integers[6]. In addition, the Giophantus encryption scheme has the ring homomorphic property described in Section 11.

Table 1 shows the difference between Giophantus and other post-quantum cryptography (PQC) candidates. In Table 1, “Linearity” indicates the linearity of the underlying problem. Giophantus provides public-key cryptosystem whose security depends on the computational hardness of solving indeterminate equations. Solving non-linear indeterminate equations is a well-known hard problem in general. In particular, it is known that there is no general solution for equations over $\mathbb{Z}$ or $F_q[t]$ and no general algorithm for solving them. Although this encryption scheme employs indeterminate equations over $F_q[t]/(t^n - 1)$, the scheme itself is potentially secure since we are able to apply non-linear equations to the scheme.

This paper is organized as follows. Section 2 gives our notation and the section 3 introduces basic mathematical definitions. In the section 4, we recall the algebraic surface encryptions, which our scheme was developed from. In section 5, we define domain parameters and propose our encryption primitive. Section 6 provides the computational assumption that makes our scheme provably secure. In the section 7, we discuss some considerable attacks against this assumption with computational experiments and the section 8 provides ap-

[6] In the paper [4], we referred to this as the **IEC** (Indeterminate Equation Cryptosystem) **encryption scheme**, but “IEC” may be confused with the standard abbreviation for the International Electrotechnical Commission, and so we adopt “Giophantus” instead.

Table 1. Comparison with other PQC candidates

Cryptosystem	Underlying problem	Linearity	Provably secure
Code Based	Decoding Problem	Linear+noise	Yes
Lattice Based	Shortest/Closest Vector Problem	Linear+noise	Yes
Multivariate	Solving Multivariate Equations	Non-linear	No
Giophantus (Present)	Solving Indeterminate Equation	Linear/Non-linear +noise	Yes

propriate parameters. The section 9 makes our primitive IND-CCA secure by applying Fujisaki-Okamoto conversion and the section 10 shows performance of our scheme. The section 11 shows that our IND-CPA primitive has homomorphic properties which will be benefit to cloud computing. We summarize the results and discuss directions for future work in Section 12.

2 Notation

The notation in this paper includes the following.

M	Plaintext in the set $\{0,1\}^k$, where k is bit length of the plaintexts. The bit length is defined in domain parameters which is described in Section 5.1.
ℓ	A small integer which is larger than 1
$(m_1 m_2 \cdots m_k)_\ell$	ℓ-ary representation of plaintext M, particularly the case $\ell = 2$, which is binary representation.
q	A prime number much larger than ℓ
F_q	The prime field identified with the set $\{0, \cdots, q-1\}$
x, y, t	Variables used for the cryptographic primitives and scheme
$F_q[t]$	Univariate polynomial ring over F_q
R_q	Quotient ring $F_q[t]/(t^n - 1)$, which is $F_q[t]$ modulo $t^n - 1$, where n is an integer larger than 1
R_ℓ	Subset of the quotient ring R_q, which consists of all univariate polynomials of t up to degree $n-1$ whose coefficients are within the range $\{0, \cdots, \ell-1\}$
$\mathbb{Z}[t]/(t^n - 1)$	Quotient ring, which is $\mathbb{Z}[t]$ modulo $t^n - 1$ where n is an integer larger than 1
n	Degree of the modulus $t^n - 1$ of the quotient ring R_q
$max(S)$	Maximum value of ordered set S. If $S = \{3, 8, -3, 4, 9\}$, then $max(S) = 9$.
$X(x,y)$	Irreducible bivariate polynomial of x and y over the ring R_q, with $X(x,y)$ an element of $R_q[x,y]$
$X(x,y) = 0$	Indeterminate equation over the ring R_q

$r(x, y)$	Random bivariate polynomial of x and y over the ring R_q, with $r(x, y)$ an element of $R_q[x, y]$
$e(x, y)$	Noise bivariate polynomial of x and y over the ring R_q, with $e(x, y)$ an element of $R_\ell[x, y]$
$m(t)$	Plaintext polynomial that embeds a plaintext M into R_ℓ
$c(x, y)$	Ciphertext polynomial over the ring R_q such that $c(x, y)$ is an element of $R_q[x, y]$
$(u_x(t), u_y(t))$	Small solution of the indeterminate equation $X(x, y) = 0$ over the ring R_q, where $u_x(t)$ and $u_y(t)$ are polynomials of t in R_ℓ and satisfy the relation $X(u_x(t), u_y(t)) = 0$
$a_{ij}(t)$	Coefficient of the monomial $x^i y^j$ belonging to the irreducible bivariate polynomial $X(x, y)$ over the ring R_q, such that $a_{ij}(t)$ is an element of R_q
$r_{ij}(t)$	Coefficient of the monomial $x^i y^j$ belonging to the random bivariate polynomial $r(x, y)$ over the ring R_q, such that $r_{ij}(t)$ is an element of R_q
$e_{ij}(t)$	Coefficient of the monomial $x^i y^j$ belonging to the noise bivariate polynomial $e(x, y)$ over the set R_ℓ, such that $e_{ij}(t)$ is an element of R_ℓ
Γ_X	Support set of the irreducible polynomial $X(x, y)$. Each element is a pair (i, j) of the exponents of $x^i y^j$, which is a non-zero monomial of $X(x, y)$ such that $\Gamma_X = \{(i, j) \in (\mathbb{N} \cup \{0\})^2 \| a_{ij}(t) \neq 0\}$.
$\#\Gamma_X$	Cardinality of the support set Γ_X
$\mathfrak{F}_{\Gamma_X}/R_q$	Set of bivariate polynomials whose support set is Γ_X over the ring R_q
Γ_r	Support set of the random polynomial $r(x, y)$. Each element is a pair (i, j) of the exponents of a non-trivial monomial $x^i y^j$.
$\#\Gamma_r$	Cardinality of the support set Γ_r
$\mathfrak{F}_{\Gamma_r}/R_q$	Set of bivariate polynomials whose support set is Γ_r over the ring R_q
Γ_e	Support set of the random polynomial $e(x, y)$. Each element is a pair (i, j) of the exponents of a non-trivial monomial $x^i y^j$.
$\#\Gamma_e$	Cardinality of the support set Γ_e
$\mathfrak{F}_{\Gamma_e}/R_\ell$	Set of bivariate polynomials whose support set is Γ_e over the ring R_ℓ
$\mathfrak{X}(\Gamma_X, \ell)/R_q$	Subset of $\mathfrak{F}_{\Gamma_X}/R_q$, consisting of all bivariate polynomials with a small zero point $(u_x(t), u_y(t))$ in R_ℓ
dX	Total degree of irreducible bivariate polynomial $X(x, y)$ such that $dX = max(\{\ i{+}j \mid X(x, y) = \sum_{(i,j)\in\Gamma_X} a_{ij}(t)x^i y^j\ \})$
dr	Total degree of random bivariate polynomial $r(x, y)$ such that $dr = max(\{\ i + j \mid r(x, y) = \sum_{(i,j)\in\Gamma_r} r_{ij}(t)x^i y^j\ \})$
$\|.\|$	Bit length of an integer, such as $\|5\| = 3$
$a\|\|b$	String concatenation of a and b.

3 Preliminaries

In this section, we introduce some basic mathematical definitions and operations needed in this paper.

3.1 Finite fields and polynomial Rings

A field is defined as a set with operations such as addition, subtraction, multiplication and division that satisfy certain rules. Typical examples of fields are the real number field $\mathbb{R}$, the rational number field $\mathbb{Q}$ and finite fields F_q. Finite fields F_q are fields with q elements, where q is a positive integer, called the order. It is well known that the order is a prime p or a prime power p^k. A prime field is defined as a finite field whose order is prime. In this paper, we focus on the case of prime fields written as sets:

$$F_q = \{0, 1, \cdots, q-1\}.$$

Its operations are described using the modulus of q as follows:

$$\begin{aligned} a+b &= a+b \mod q, \\ a-b &= a-b \mod q, \\ a \cdot b &= a \cdot b \mod q, \\ a/b &= a \cdot b^{-1} \mod q, \end{aligned} \tag{1}$$

where b^{-1} satisfies the condition $b \cdot b^{-1} = 1 \mod q$.

Example 1. The prime field $F_5 = \{0, 1, 2, 3, 4\}$ can be equipped with operations modulo 5, such as

$$1+2=3, \quad 2+4=1, \quad 3-1=2, \quad 2-3=4,$$

$$2 \cdot 2 = 4, \quad 2 \cdot 3 = 1, \quad 2/3 = 2 \cdot 3^{-1} = 2 \cdot 2 = 4.$$

Let R be a ring. A univariate polynomial ring is a set defined as

$$R[t] = \{c_0 + c_1 t + \cdots + c_n t^n \mid c_i \in R \ (0 \le i \le n) \ n \in \mathbb{N}\}, \tag{2}$$

where t is a variable and c_i is the coefficient of the monomial $c_i t^i$. Univariate polynomials $f(t)$ and $g(t)$ can be described as

$$\begin{aligned} f(t) &= a_0 + a_1 t + \cdots + a_n t^n, \\ g(t) &= b_0 + b_1 t + \cdots + b_n t^n, \end{aligned} \tag{3}$$

where a_i and b_i are elements of R. We note that neither $a_n = 0$ nor $b_n = 0$ is assumed in the expression of (3) above.

$R[t]$ is a ring since addition and multiplication are defined as follows:

$$\begin{aligned} f+g &= a_0 + b_0 + (a_1 + b_1)t + \cdots + (a_n + b_n)t^n, \\ f \cdot g &= a_0 \cdot b_0 + (a_1 \cdot b_0 + a_0 \cdot b_1)t + \cdots + (a_n \cdot b_n)t^{2n}. \end{aligned} \tag{4}$$

Though an inverse of addition can be defined as

$$-f = -a_0 - a_1 t - \cdots - a_n t^n \ ,$$

an inverse of multiplication can be defined if and only if $f(t)$ is a non-zero constant, such as $f(t) = a_0$.

Example 2. Let us consider a univariate polynomial in $F_5[t]$ and set $f(t) = 2 + 3t + 4t^2$ and $g(t) = 4 + t + 3t^2$. Then, $f(t) + g(t) = 1 + 4t + 2t^2$, $f(t) \cdot g(t) = 2t^4 + 3t^3 + 4t + 3$, and $-f(t) = 3 + 2t + t^2$.

$$F_5[t] = \{c_0 + c_1 t + \cdots + c_n t^n \mid c_i \in F_5 \ (0 \leq i \leq n) \ n \in \mathbb{N}\} \ . \tag{5}$$

If a polynomial is written in $f(t) = \sum_{i=0}^{n} c_i t^i$ such that the coefficient $c_n \neq 0$ then we define n to be the degree of f. Thus, the degree of f is the maximum integer n such that $c_n \neq 0$. We denote this by $\deg f = n$. In the example of $f(t)$ and $g(t)$ above,

$$\deg(f) = \deg(g) = 2, \ \ \deg(f(t) \cdot g(t)) = 4 \, .$$

A bivariate polynomial ring is a set defined as

$$\begin{aligned} R[x,y] = \{&c_{n0}x^n + c_{(n-1)1}x^{n-1}y + \cdots + c_{0n}y^n + \cdots + c_{10}x + c_{01}y + c_{00} \\ &\mid c_{ij} \in R \ (0 \leq i, j \leq n) \ n \in \mathbb{N}\}, \end{aligned} \tag{6}$$

where x and y are variables and c_{ij} are coefficients of the monomial $c_{ij}x^i y^j$.

Set $f(x,y)$ and $g(x,y)$ as follows:

$$\begin{aligned} f(x,y) &= \textstyle\sum_{i=j=1}^{n} a_{ij} x^i y^j \\ &= a_{n0}x^n + a_{(n-1)1}x^{n-1}y + \cdots + a_{0n}y^n + \cdots + a_{10}x + a_{01}y + a_{00}, \\ g(x,y) &= \textstyle\sum_{i=j=1}^{n} b_{ij} x^i y^j \\ &= b_{n0}x^n + b_{(n-1)1}x^{n-1}y + \cdots + b_{0n}y^n + \cdots + b_{10}x + b_{01}y + b_{00} \ , \end{aligned} \tag{7}$$

where a_{ij} and b_{ij} are elements of R. Then we define addition and multiplication as follows:

$$\begin{aligned} f + g &= \textstyle\sum_{i=j=0}^{n} (a_{ij} + b_{ij}) x^i y^j \\ &= (a_{n0} + b_{n0})x^n + (a_{(n-1)1} + b_{(n-1)1})x^{n-1}y + \cdots + (a_{10} + b_{10})x \\ &\quad + (a_{01} + b_{01})y + a_{00} + b_{00}, \\ f \cdot g &= \textstyle\sum_{i_1+j_1=i_2+j_2=0}^{n} (a_{i_1 j_1} b_{i_2 j_2}) x^i y^j \\ &= (a_{n0}b_{n0})x^{2n} + (a_{n0}b_{(n-1)1} + a_{(n-1)1}b_{n0})x^{2n-1}y + \cdots \\ &\quad + (a_{01}b_{00} + a_{00}b_{01})y + a_{00}b_{00}. \end{aligned} \tag{8}$$

An inverse of addition can be defined as

$$-f = -a_{n0}x^n - a_{(n-1)1}x^{n-1}y - \cdots - a_{0n}y^n - \cdots - a_{10}x - a_{01}y - a_{00} \, .$$

However, an inverse of multiplication does not exist in general.

Example 3. In the case of $F_5[x, y]$, set

$$\begin{aligned} f(x,y) &= 2x^2 + 3xy + y^2 + 3x + 3y + 4, \\ g(x,y) &= x^2 + 2xy + 3y^2 + x + 3y + 3\,, \end{aligned} \tag{9}$$

and then $f(x, y) + g(x, y) = 3x^2 + 4y^2 + 4x + y + 2$,

$$\begin{aligned} f(x,y) \cdot g(x,y) = {} & 2\,x^4 + 2\,x^3y + 3\,x^2y^2 + xy^3 + 3\,y^4 + 3\,x^2y + 2\,y^3 + 3\,x^2 \\ & +4\,xy + 4\,y^2 + 3\,x + y + 2 \end{aligned}$$

and $-f(x, y) = 3x^2 + 2xy + 4y^2 + 2x + 2y + 1$.

Setting the bivariate polynomial $f(x, y) = \sum_{i=j=0}^{n} c_{ij}x^iy^j$, the total degree of f, denoted $\deg f$, can be defined as

$$\deg f := max(\{i + j | c_{ij} \neq 0\})\,.$$

We can determine the degrees for $f(x, y)$ and $g(x, y)$, described above, as follows.

$$\deg(f(x, y)) = \deg(g(x, y)) = 2, \;\; \deg(f(x, y) \cdot g(x, y)) = 4\,.$$

3.2 The quotient ring R_q

The ring R_q is defined as the quotient ring of $F_q[t]$ modulo $t^n - 1$. Elements of R_q are polynomials over F_q with degree at most $n - 1$ (since t^n is equivalent to 1).

We can represent $a \in R_q$ as a vector $(a_0, a_1, \cdots, a_{n-2}, a_{n-1})$ representing

$$a = a_0 + a_1t + \cdots + a_{n-2}t^{n-2} + a_{n-1}t^{n-1}$$

on F_q. When elements $b, c \in R_q$ are represented in the same manner as a, we can express $ab + c$ as

$$\begin{pmatrix} b_0 \; b_1 \cdots b_{n-2} \; b_{n-1} \end{pmatrix} \begin{pmatrix} a_0 & a_1 & \cdots & a_{n-2} & a_{n-1} \\ a_{n-1} & a_0 & \cdots & a_{n-3} & a_{n-2} \\ a_{n-2} & a_{n-1} & \cdots & a_{n-4} & a_{n-3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_1 & a_{n-1} & \cdots & a_{n-1} & a_0 \end{pmatrix} + \begin{pmatrix} c_0 \; c_1 \cdots c_{n-2} \; c_{n-1} \end{pmatrix} \tag{10}$$

on F_q.

Using this expression, the relation $ab + c = d$ can be described as

$$\boldsymbol{b}A + \boldsymbol{c} = \boldsymbol{d}\,,$$

where vectors $\boldsymbol{b}$ and $\boldsymbol{c}$ correspond to b and c, respectively, and A is a matrix corresponding to a. The vector $\boldsymbol{d}$ corresponds to the result of $ab + c$.

3.3 Monomial order

To describe a detailed specification of the proposal, we need to introduce the monomial order of polynomials, which defines the order of calculation. First, we define an exponent vector $\alpha = (i, j) \in \mathbb{Z}^2_{\geq 0}$ of monomial $x^i y^j$, and then we denote a monomial $x^i y^j$ as x^α.

Example 4. The exponent vectors of monomials $3x^2y^3$ and $4x^3$ in $F_q[x, y]$ are $(2, 3)$ and $(3, 0)$ respectively.

We define the monomial ordering $x^\alpha > x^\beta$ as follows.

Definition 1. *A monomial ordering on bivariate polynomial ring $F_q[x, y]$ is any relation $>$ on the set of monomials in $F_q[x, y]$ or $\mathbb{Z}^2_{\geq 0}$ satisfying:*

1. *$>$ is a total ordering such that any pair of monomials α and β satisfies exactly one of the relations $\alpha < \beta$, $\alpha = \beta$, and $\alpha > \beta$.*
2. *$>$ is compatible with multiplication in $F_q[x, y]$. If $\alpha > \beta$ and there is some $\gamma \in \mathbb{Z}^2_{\geq 0}$ then $\alpha + \gamma > \beta + \gamma$ since the relation $x^\alpha x^\gamma > x^\beta x^\gamma$ is satisfied.*
3. *$>$ induces a well ordering, such that there is a minimum element in any non-empty subset of $\mathbb{Z}^2_{\geq 0}$ or monomial set.*

Lexicographic ordering is an example of monomial ordering satisfying these rules. It is defined as follows.

Definition 2. *For any $\alpha = (\alpha_1, \alpha_2) \in \mathbb{Z}^2_{\geq 0}$ and $\beta = (\beta_1, \beta_2) \in \mathbb{Z}^2_{\geq 0}$, the relation $\alpha >_{lex} \beta$ (resp., $\alpha <_{lex} \beta$) holds when the leftmost non-zero entry of the difference of the exponent vectors $\alpha - \beta$ is positive (resp., negative). We write $x^\alpha >_{lex} x^\beta$ if $\alpha >_{lex} \beta$ and analogously for $<_{lex}$.*

For example, $(2, 1) >_{lex} (1, 2)$ since the difference of the exponent vectors $\alpha - \beta = (1, -1)$. Similarly, $(2, 1) <_{lex} (2, 2)$ since $\alpha - \beta = (0, -1)$, and the leftmost non-zero entry is negative.

In this paper, we employ the graded lexicographic order, which is defined as follows.

Definition 3. *Let x^α and x^β be monomials in $F_q[x, y]$. We define $x^\alpha <_{grlex} x^\beta$ if $\alpha_1 + \alpha_2 > \beta_1 + \beta_2$, or if $\alpha_1 + \alpha_2 = \beta_1 + \beta_2$ and in the difference vector $\alpha - \beta$, the leftmost non-zero entry is positive.*

For example, we have $(0, 2) >_{grlex} (1, 0)$ since $\alpha_1 + \alpha_2 = 2 > 1 = \beta_1 + \beta_2$. In the case of $(1, 1) >_{grlex} (0, 2)$, we have $\alpha_1 + \alpha_2 = 2 = \beta_1 + \beta_2$ and $\alpha - \beta = (1, -1)$, the leftmost non-zero entry is positive.

3.4 Support set of a polynomial

Let's R be a set. Let's $X(x, y)$ be a bivariate polynomial over R written as $\sum_{(i,j)\in(\mathbb{N}\cup\{0\})^2} a_{ij} x^i y^j$, where a_{ij} is an element of R. Then we can define a support set of a polynomial $X(x, y)$ over the set R as

$$\mathit{\Gamma}_X = \{(i, j) \in (\mathbb{N} \cup \{0\})^2 \mid a_{ij} \neq 0 \ s.t. \ X(x, y) = \sum_{(i,j)\in(\mathbb{N}\cup\{0\})^2} a_{ij} x^i y^j\}$$

The support set Γ_X specifies the monomials which belongs to a bivariate polynomial $X(x, y)$. In addition, we can define the support set Γ_{Xr} for given support sets Γ_X and Γ_r as follows.

$$\Gamma_{Xr} = \\ \{(i, j) \in (\mathbb{N} \cup \{0\})^2 \mid i = i_1 + i_2, j = j_1 + j_2 \ s.t.\ (i_1, j_1) \in \Gamma_X, (i_2, j_2) \in \Gamma_r\} \tag{11}$$

The Γ_{Xr} is a support set of the polynomial Xr's, where X and r are polynomials whose support sets are Γ_X and Γ_r respectively.

The bivariate polynomial set of all polynomials whose support set is a subset of Γ_X is defined as

$$\mathfrak{F}_{\Gamma_X}/R = \{f(x, y) \in R[x, y] \mid a_{ij} \neq 0 \Rightarrow (i, j) \in \Gamma_X\} \ . \tag{12}$$

4 Design concept

4.1 Algebraic Surface Cryptosystem

The ASC was first announced in 2006 by K. Akiyama and Y. Goto [2]. The algebraic surfaces are defined as a solution space of a three-variable polynomial equation $X(x, y, t) = 0$ over a field K. The security of ASC depends on the section-finding problem, defined as follows.

Definition 4. (Section-finding problem) If $X(x, y, t) = 0$ is an algebraic surface over a field K, then the problem of finding a parameterized curve $(x, y, t) = (u_x(t), u_y(t), t)$ on X is called the *section-finding problem* on X.

A section can be considered as a solution of $X(x, y) = 0$, which is an indeterminate equation over the ring $K[t]$.

The problem of solving indeterminate equations over an arbitrary ring or field is known to be hard. For example, the class of indeterminate equations over the integer ring $\mathbb{Z}$, called Diophantine equations, is undecidable (Hilbert's 10th problem). Being "undecidable" means that there is no general algorithm to solve such indeterminate equations. The section-finding problem has been proven to be undecidable [17].

We recall the method of algebraic surface encryption to see the conceptual design for the scheme described in this paper. First, the simplest ASC can be described as

$$c(x, y) = m(x, y) + X(x, y)r(x, y),$$

where $X(x, y)$ is the public key, which defines an algebraic surface with a section. The polynomials $c(x, y)$ and $r(x, y)$ are a ciphertext polynomial and a random polynomial, respectively. The polynomial $m(x, y)$ is a plaintext polynomial in which a plaintext message is embedded. In the decryption phase, we substitute the secret key (a section of $X(x, y)$) into $c(x, y)$. By the relation $X(u_x(t), u_y(t)) = 0$, we obtain

$$c(u_x(t), u_y(t)) = m(u_x(t), u_y(t)).$$

From the polynomial $m(u_x(t), u_y(t))$, we can recover the plaintext message as follows. We can describe $m(x, y)$ as

$$m(x, y) = \sum_{(i,j,k)\in\Gamma_m} m_{ijk} x^i y^j t^k,$$

where each m_{ijk} is a variable, and substitute the section into $m(x, y)$. We thus obtain

$$m(u_x(t), u_y(t)) = \sum_{(i,j,k)\in\Gamma_m} m_{ijk} u_x(t)^i u_y(t)^j t^k.$$

Comparing the coefficient of t, the simultaneous linear equations containing m_{ijk} are constructed. When the number of variables is less than or equal to the number of equations, we can detect the correct plaintext message by solving the equations.

However, there exists an attack to break the scheme. We can expand the cipher polynomial $c(x, y)$ to

$$c(x, y) = \sum_{(i,j,k)\in\Gamma_m} m_{ijk} x^i y^j t^k + \left(\sum_{(i,j,k)\in\Gamma_X} a_{ijk} x^i y^j t^k \right) \left(\sum_{(i,j,k)\in\Gamma_r} r_{ijk} x^i y^j t^k \right), \tag{13}$$

where Γ_m,Γ_X, and Γ_r are given as parameters, and the values a_{ijk} are the given coefficients of the public key X. Each m_{ijk} and r_{ijk} is a variable. Comparing the coefficients of the monomials, we obtain the simultaneous linear equations having the variables m_{ijk} and r_{ijk}. For decryption, the relation

$$\#\Gamma_m + \#\Gamma_r < \#\Gamma_{Xr}$$

is required. However, in this case, the equations have unique solution with high probability. We refer to this type of attack as the **Linear Algebra attack**.

To avoid the attack, K. Akiyama, Y. Goto and H. Miyake constructed the latest ASC scheme in 2009 [3]. From the cryptographic point of view, the ciphertext is equivalent to

$$c(x, y) = m(x, y)s(x, y) + X(x, y)r(x, y). \tag{14}$$

Here, $s(x, y)$ is employed as another random polynomial and the term product $m(x, y)s(x, y)$ equals $X(x, y)r(x, y)$ (with $\Gamma_{ms} = \Gamma_{Xr}$). To decrypt the ciphertext, we have to divide $m(u_x(t), u_y(t))s(u_x(t), u_y(t))$ into $m(u_x(t), u_y(t))$ and $s(u_x(t), u_y(t))$ by factoring. Since polynomial factoring is computationally easy via the Berlekamp method, we can obtain $m(u_x(t), u_y(t))$ as a factor. The plaintext is then recovered from $m(u_x(t), u_y(t))$ in the same way as in the previous scheme.

Applying the Linear Algebra attack to this scheme, we need to consider $m(x, y)s(x, y)$ as a single polynomial $g(x, y)$, since quadratic equations are derived from the variables m_{ijk} and s_{ijk}. Therefore, the number of variables,

$\#\Gamma_r + \#\Gamma_{Xr}$, is greater than the number of equations, $\#\Gamma_{Xr}$, and so the Linear Algebra attack does not work.

Unfortunately, this scheme was also broken by the **ideal decomposition attack**, which was described by Faugere et al. [19]. They found that the ideal $(c(x,y), X(x,y))$ can be decomposed into $(m(x,y), X(x,y))$ and $(s(x,y), X(x,y))$ by calculating the resultant $Res_x(c(x,y), X(x,y))$ and $Res_y(c(x,y), X(x,y))$. The plaintext message $m(x,y)$ is then recovered by solving the linear equations.

The proposed primitive avoids both attacks. Our idea is to apply for the ℓ-polynomial structure employed in NTRU encryption. The ciphertext is

$$c(x,y) = m(x,y) + X(x,y)r(x,y) + \ell \cdot e(x,y),$$

where $e(x,y)$ is a random polynomial whose coefficients are small. The polynomial $e(x,y)$ works as a noise factor in the cipher, and we claim the condition

$$\#\Gamma_e = \#\Gamma_{Xr}$$

for resistance against the Linear Algebra attack. Needing the smallest solution of $X(x,y)$ to decrypt the message ensures this.

5 Our Proposed Encryption Scheme

This section provides an overview of the proposed encryption scheme.

5.1 Domain parameters

We introduce parameters for the proposed scheme to be input to the key generation algorithm. Appropriate parameter settings are discussed in Section 8.

- ℓ: A small integer which is larger than 1.
- q: A prime which is cardinality of prime field F_q and is much larger than ℓ.
- n: Degree of the modulus polynomial of the quotient ring $R_q(= F_q[t]/(t^n - 1))$. The n should be prime for the security reason.
- dX: Total degree of the irreducible bivariate polynomial $X(x,y)$
- dr: Total degree of the random bivariate polynomial $r(x,y)$
- $mlen$ Length of the message M

The relation between ℓ and q is a critical condition for decryption. We require the condition

$$q > \ell - 1 + \ell \sum_{k=0}^{dX+dr} (k+1)n^k(\ell-1)^{k+1} \tag{15}$$

to decrypt any ciphertext encrypted by the proposed encryption primitive.

The support set of the irreducible polynomial $X(x,y)$ with total degree dX is defined such that

$$\Gamma_X = \{(i,j) \in (\mathbb{N} \cup \{0\})^2 \ \ | \ \ 0 \le i, j, i+j \le dX\}$$

with graded lexicographic order. If dX is equal to 2, then

$$\Gamma_X = \{(2,0),(1,1),(0,2),(1,0),(0,1),(0,0)\},$$

whose elements correspond to the monomials x^2,xy,y^2,x,y, and 1, in that order, and the monomial order is called the graded lexicographic order.

The support set of the random polynomial $r(x,y)$ with total degree dr is also defined such that

$$\Gamma_r = \{(i,j) \in (\mathbb{N}\cup\{0\})^2 \mid 0 \le i,j,i+j \le dr\}$$

with graded lexicographic order. Since the total degree of the noise polynomial $e(x,y)$ is defined to be $dX + dr$, the Support set of the noise polynomial $e(x,y)$ is

$$\Gamma_e = \{(i,j) \in (\mathbb{N}\cup\{0\})^2 \mid 0 \le i,j,i+j \le dX+dr\}$$

with graded lexicographic order. If $dX = dr = 2$, then

$$\Gamma_e = \{(4,0),(3,1),(2,2),(1,3),(0,4),(3,0),(2,1),(1,2),(0,3),(2,0), \\ (1,1),(0,2),(1,0),(0,1),(0,0)\},$$

whose elements correspond to the monomials x^4,x^3y,x^2y^2,xy^3,y^4,x^2y,xy^2,y^3,x^2, xy,y^2,x,y, and 1, in that order.

5.2 Key Generation

The secret key is a small (not necessarily smallest) solution of the indeterminate equation $X(x,y)=0$:

$$(x,y) = (u_x(t), u_y(t)), \quad u_x(t), u_y(t) \in R_\ell, \tag{16}$$

where $\deg u_x(t) = \deg u_y(t) = n-1$. Note that ℓ is much smaller than q, and thus we call $(u_x(t), u_y(t))$ a small solution. The public key is the indeterminate equation $X(x,y)=0$, which is irreducible and has the small solution $(u_x(t), u_y(t))$:

$$X(x,y) = \sum_{(i,j)\in\Gamma_X} a_{ij}(t)x^iy^j \ , \tag{17}$$

where $a_{ij}(t) \in R_q$.

The key generation algorithm takes the parameters ℓ,q,n,dX, and dr as parameters, and is defined in Section 5.1. The secret key is generated as degree $n-1$ random polynomials $u_x(t), u_y(t) (\in R_\ell)$. The indeterminate equation $X(x,y)=0$ is constructed according to the following procedure.

1. Generate a degree dX support set Γ_X with graded lexicographic order.
2. Choose a coefficient of each monomial (except the constant term) as follows.
   (a) Set $X(x,y)=0$
   (b) For each element (i,j) in $\Gamma_X - \{(0,0)\}$
      i. Choose a coefficient $a_{ij}(t)$ whose degree is $n-1$, uniformly at random from the set R_q
      ii. Set $X(x,y) = X(x,y) + a_{ij}(t)x^iy^j$
3. Calculate the constant term $a_{00}(t)$ as
   $a_{00}(t) = -\sum_{(i,j)\in\Gamma_X-\{(0,0)\}} a_{ij}(t)u_x(t)^iu_y(t)^j (\in R_q)$
4. Confirm the polynomial $X(x,y)$ is irreducible; if not, return to step 2a.

5.3 Encryption

1. Embed a plaintext M into the coefficients of the plaintext polynomial $m(t)(\in R_\ell)$ whose degree is $n-1$. As an example, in the case of $\ell = 4, n = 3$, a plaintext $M = (312)_4$ can be embedded such as $m(t) = 3t^2 + t + 2$.
2. Generate a support set Γ_r of degree dr with graded lexicographic order
3. Create a random polynomial $r(x, y)$ as follows:
   (a) Set $r = 0$
   (b) For each (i, j) in Γ_r
      i. Choose a coefficient $r_{ij}(t)$ uniformly at random from the set R_q
      ii. Set $r(x, y) = r(x, y) + r_{ij}(t)x^i y^j$
4. Generate a support set Γ_e of degree $dX + dr$ with graded lexicographic order
5. Create a noise polynomial $e(x, y)$ as follows:
   (a) Set $e(x, y) = 0$
   (b) For each (i, j) in Γ_e
      i. Choose a coefficient $e_{ij}(t)$ uniformly at random from the set R_ℓ
      ii. Set $e(x, y) = e(x, y) + e_{ij}(t)x^i y^j$
6. Construct the cipher polynomial $c(x, y)$ as

$$c(x, y) = m(t) + X(x, y)r(x, y) + \ell \cdot e(x, y) \tag{18}$$

5.4 Decryption

1. Substitute the secret key that is a small solution $(u_x(t), u_y(t))$ over R_q of $X(x, y) = 0$ into $c(x, y)$:

$$c(u_x(t), u_y(t)) = m(t) + \ell \cdot e(u_x(t), u_y(t)) \tag{19}$$

   When the parameters ℓ and q satisfy the relation described above (15), each coefficient of $m(t) + \ell \cdot e(u_x(t), u_y(t)) \in \mathbb{Z}/(t^n - 1)$ is within the range from 0 to $q - 1$. Theorem 1 gives a proof of this fact.
2. Extract $m(t)$ from $c(u_x(t), u_y(t))$ as

$$c(u_x(t), u_y(t)) \pmod{\ell} = m(t),$$

   where we consider $c(u_x(t), u_y(t))$ as an element of $\mathbb{Z}[t]$
3. Recover the plaintext M from the coefficients of $m(t)$.

Theorem 1. *Let a ciphertext polynomial $c(x, y)(\in R_q[x, y])$ encrypt a plaintext polynomial $m(t)(\in R_\ell)$ with a public key $X(x, y)$ and public parameters (n, ℓ, q, dX, dr), applying the encryption algorithm in the section 5.3. The plaintext polynomial $m(t)$ can be recovered from the ciphertext $c(x, y)$ with a corresponding secret key $(u_x(t), u_y(t))$ and public parameters (n, ℓ, q, dX) by applying the decryption algorithm in the section 5.4.*

Proof. Since a secret key $(u_x(t), u_y(t))$ is a solution of the equation $X(x, y) = 0$, we obtain

$$c(u_x(t), u_y(t)) = m(t) + \ell \cdot e(u_x(t), u_y(t)) \pmod{\ell},$$

where the calculation is in the ring $R_q[x, y]$.

Take $m(t)+\ell\cdot e(u_x(t), u_y(t))$ of R_q as a univariate polynomial over the integers $\mathbb{Z}$, where the coefficients are integers within the range 0 to $q-1$. Now we denote by $MC(f(t))$ the maximum coefficient of a univariate polynomial $f(t)$ over the integer $\mathbb{Z}$. If the condition

$$MC(m(t) + \ell \cdot e(u_x(t), u_y(t))) < q \tag{20}$$

is satisfied in the univariate polynomial ring $\mathbb{Z}[t]$ for any possible $m(t), e(x, y)$, $(u_x(t), u_y(t)), \ell$, then the conclusion

$$m(t) + \ell \cdot e(u_x(t), u_y(t)) \pmod{\ell} = m(t)$$

follows. Here, $m(t)$ is an element of R_ℓ whose coefficients are restricted to the range 0 to $\ell - 1$.

To see the relation (20), we assume the coefficients of the polynomials $u_x(t), u_y(t)$ are maximum, such as

$$u_x(t) = u_y(t) = (\ell - 1)(t^{n-1} + t^{n-2} + \cdots + t + 1).$$

We can see $(t^{n-1}+t^{n-2}+\cdots+t+1)^k = n^{k-1}(t^{n-1}+t^{n-2}+\cdots+t+1)$ for any positive integer k since the multiples have to be reduced by $t^n - 1$. Then

$$u_x(t)^k = u_y(t)^k = (\ell - 1)^k \cdot n^{k-1}(t^{n-1} + t^{n-2} + \cdots + t + 1),$$

The support set $\mathit{\Gamma}_e$ is

$$\mathit{\Gamma}_e = \{(i, j) \in (\mathbb{N} \cup \{0\})^2 | 0 \le i, j, i + j \le dX + dr\}.$$

Since there are ${}_2H_k$ degree-k elements in $\mathit{\Gamma}_e$, the value of $MC(e(u_x(t), u_y(t))$ is as follows:

$$\begin{aligned}
MC(e(u_x(t), u_(t)) &= MC(\textstyle\sum_{(i,j)\in\mathit{\Gamma}_e} e_{ij}(t)u_x(t)^i u_y(t)^j) \\
&\le MC(\textstyle\sum_{(i,j)\in\mathit{\Gamma}_e}(\ell-1)(t^{n-1}+t^{n-2}+\cdots+t+1)u_x(t)^i u_y(t)^j) \\
&= (\ell-1)\textstyle\sum_{k=0}^{dX+dr} {}_2H_k n^k \dot{(}\ell-1)^k \\
&= \textstyle\sum_{k=0}^{dX+dr} {}_{k+1}C_k n^k \dot{(}\ell-1)^{k+1} \\
&= \textstyle\sum_{k=0}^{dX+dr} (k+1)n^k \dot{(}\ell-1)^{k+1}.
\end{aligned}$$

So, we obtain the relation

$$MC(m(t) + \ell \cdot e(u_x(t), u_(t)) \le \ell - 1 + \ell \sum_{k=0}^{dX+dr} (k+1)n^k \dot{(}\ell - 1)^{k+1}.$$

The condition (20) is always satisfied since $q > \ell - 1 + \ell \sum_{k=0}^{dX+dr}(k+1)n^k \dot{(}\ell - 1)^{k+1}$.

6 Security assumption and proof for primitives (IND-CPA)

In this section, we introduce a computational assumption and discuss some possible attacks under this assumption, based on the attacks for ASCs.

6.1 The smallest-solution problem

Let us express the solution $u = (u_x(t), u_y(t))$ $(\in (\mathbb{Z}_q[t]/(t^n - 1))^2)$ of an indeterminate equation as

$$u_x(t) = \sum_{i=0}^{n-1} \alpha_i t^i, \quad u_y(t) = \sum_{i=0}^{n-1} \beta_i t^i.$$

The norm of the solution is defined as follows.

$$Norm(u) = \max(\{\alpha_i, \beta_i \in \mathbb{Z}_q^+ \mid 0 \leq i \leq n-1\})$$

The security of our system depends on the smallest-solution problem, defined as follows.

Definition 5. (Smallest-solution Problem) If $X(x, y) = 0$ is an indeterminate equation over the ring $\mathbb{Z}_q[t]/(t^n - 1)$, then the problem of finding the solution $(x, y) = (u_x(t), u_y(t))$ on $\mathbb{Z}_q[t]/(t^n - 1)$ with the smallest norm is called the *smallest-solution problem* on X.

Approximate lattice reduction algorithms cannot be directly applied to solving the problem because the solution space is non-linear.

6.2 Security assumption

Polynomials over $\mathbb{Z}_q$ whose coefficients are in the range 0 to $p-1$ are called size-ℓ polynomials. If a polynomial is size ℓ, this means that its coefficients are much smaller than those of an ordinary polynomial, since ℓ is much smaller than q. We define the set of polynomials that have zero points in size ℓ as follows:

$$\mathfrak{X}(\Gamma_X, \ell)/R_q = \{X \in \mathfrak{F}_{\Gamma_X}/R_q \mid \exists u_x(t), u_y(t) \in R_\ell \;\; X(u_x(t), u_y(t)) = 0\}.$$

Given sets of polynomials, such as $\mathfrak{X}(\Gamma_X, \ell)/R_q$, $\mathfrak{F}_{\Gamma_r}/R_q$, and $\mathfrak{F}_{\Gamma_{Xr}}/R_\ell$, that satisfy the condition

$$(0,0) \in \Gamma_X, (0,0) \in \Gamma_r,$$

we define the decision problem as follows.

Definition 6. *(IE-LWE problem) Writing the sets U_X and T_X as*

$$U_X = \mathfrak{X}(\Gamma_X, \ell)/R_q \times \mathfrak{F}_{\Gamma_{Xr}}/R_q, \tag{21}$$

$$T_X = \{(X, Xr + e) | X \in \mathfrak{X}(\Gamma_X, \ell)/R_q, r \in \mathfrak{F}_{\Gamma_r}/R_q, e \in \mathfrak{F}_{\Gamma_{Xr}}/R_\ell\}, \tag{22}$$

the IE-LWE problem is to distinguish the multivariate polynomials chosen from a "noisy" set T_X of polynomials or from a set $U_X - T_X$, where T_X is a subset of U_X.

We define the IE-LWE assumption.

Definition 7. *(IE-LWE assumption) The IE-LWE assumption is the assumption that the advantage*

$$Adv_{\mathfrak{B}}^{\text{IE-LWE}}(k) :=$$

$$\left| \begin{array}{l} Pr\left[\mathfrak{B}(\ell, q, n, \Gamma_r, \Gamma_X, X, Y) \to 1 \left| \begin{array}{l} (\ell, q, n, \Gamma_X, \Gamma_r, X) \stackrel{R}{\leftarrow} GenG(1^k); \\ r \stackrel{U}{\leftarrow} \mathfrak{F}_{\Gamma_r}/R_q; e \stackrel{U}{\leftarrow} \mathfrak{F}_{\Gamma_{Xr}}/R_\ell; \\ Y := Xr + e \end{array} \right. \right] \\ -Pr\left[\mathfrak{B}(\ell, q, n, \Gamma_r, \Gamma_X, X, Y) \to 1 \left| \begin{array}{l} (\ell, q, n, \Gamma_X, \Gamma_r, X) \stackrel{R}{\leftarrow} GenG(1^k); \\ \\ Y \stackrel{U}{\leftarrow} \mathfrak{F}_{\Gamma_{Xr}}/R_q \end{array} \right. \right] \end{array} \right| \tag{23}$$

is negligible, where the function $GenG(1^k)$ outputs the domain parameters (i.e., ℓ,q,n,Γ_X, and Γ_r) from the security parameter k and creates X from these domain parameters by the key generation algorithm in the section 5.2. In other words,

$$Adv_{\mathfrak{B}}^{\text{IE-LWE}}(k) < \epsilon(k),$$

where $\epsilon(k)$ is a negligible function in the security parameter k.

IE-LWE is an extended variation of R-LWE$^{\times}_{\text{HNF}}$, which is one of the variants of R-LWE defined by the polynomial ring R_q. This is claimed by a provably secure NTRU modification [48] and can be reduced to the shortest-vector problem of the lattice derived from R_q. In this paper, we extend R-LWE$^{\times}_{\text{HNF}}$ to the multivariate polynomial ring $R_q[x, y]$ so that the dimension of the lattice is larger than that of the lattice derived from R_q.

Theorem 2. *Under the IE-LWE assumption, the Giophantus encryption scheme $\Sigma = (Gen, Enc, Dec)$ is secure in the sense of IND-CPA. Specifically, if there is an adversary that runs in polynomial time and breaks the Giophantus encryption scheme Σ in the sense of IND-CPA, then there exists an algorithm $\mathfrak{B}$ that solves the IE-LWE problem in probabilistic polynomial time. Moreover, the following relation holds:*

$$Adv_{\Sigma,\mathfrak{A}}^{\text{IND-CPA}}(k) = 2 \cdot Adv_{\mathfrak{B}}^{\text{IE-LWE}}(k).$$

Proof. Assume that Σ is not secure in the sense of IND-CPA. Then, there exists an adversary $\mathfrak{A}$ who breaks Σ in polynomial time with non-negligible advantage

$$Adv_{\Sigma,\mathfrak{A}}^{\text{IND-CPA}}(k) \geq \epsilon(k),$$

where k is a security parameter. By using $\mathfrak{A}$, we construct an algorithm $\mathfrak{B}$ solving the IE-LWE problem in probabilistic polynomial time as follows. Without loss of generality, we assume $\mathfrak{B}$ outputs 1 when it decides that the input is sampled from T_X, and otherwise outputs 0.

Assume an oracle $\mathcal{O}$ that picks set $S \hookleftarrow U(\{T_X, U_X - T_X\})$ and samples from the set of S uniformly at random. Algorithm $\mathfrak{B}$ first calls $\mathcal{O}$ to get a sample $(X^{'}(x,y), C^{'}(x,y))$ from S. Then, the algorithm runs $\mathfrak{A}$ with the public key $X(x,y)(=\ell X^{'}(x,y) \in \mathfrak{X}(\Gamma_X,\ell)/R_q)$. Here, $X(x,y)$ is chosen uniformly at random from $\mathfrak{X}(\Gamma_X,\ell)/R_q$ since the map $X'(x,y) \rightarrow \ell X^{'}(x,y)$ is invertible due to the invertibility of ℓ modulo q.

When $\mathfrak{A}$ outputs challenge messages $m_0(t), m_1(t) \in R_\ell$, the algorithm $\mathfrak{B}$ picks b either 0 or 1 uniformly at random, computes the challenge ciphertext $c(x,y) = \ell \cdot C^{'}(x,y) + m_b(t) \in \mathfrak{F}_{\Gamma_e}/R_q$, and returns $c(x,y)$ to $\mathfrak{A}$. Finally, when $\mathfrak{A}$ outputs its guess $b^{'}$ for b, the algorithm $\mathfrak{B}$ outputs 1 if $b^{'} = b$ and 0 otherwise. Here, $c(x,y)$ is calculated as follows.

$$c(x,y) = \ell \cdot C^{'}(x,y) + m_b(t) = m_b(t) + X(x,y)r(x,y) + \ell \cdot e(x,y).$$

If the sample $(X^{'}(x,y), C^{'}(x,y))$ is from T_X, then it is impossible to distinguish $c(x,y)$ from an element chosen from the ciphertext space uniformly at randombecause $r(x,y)$, and $e(x,y)$ are chosen from $\mathfrak{F}_{\Gamma_r}/R_q$ and $\mathfrak{F}_{\Gamma_e}/R_\ell$, respectively, uniformly randomly. If the algorithm $\mathfrak{A}$ outputs $b^{'} = b$ with non-negligible advantage $Adv^{\text{IND-CPA}}_{\Sigma,\mathfrak{A}}(k)$, then we can calculate $Adv^{\text{IND-CPA}}_{\Sigma,\mathfrak{A}}(k)$ as follows.

$$\begin{aligned}
&Adv^{\text{IND-CPA}}_{\Sigma,\mathfrak{A}}(k)\\
&= |Pr[b = b^{'}|(X^{'}(x,y), C^{'}(x,y)) \overset{U}{\leftarrow} T_X] - Pr[b \neq b^{'}|(X^{'}(x,y), C^{'}(x,y)) \overset{U}{\leftarrow} T_X]|\\
&= |Pr[\mathfrak{B}(X^{'}(x,y), C^{'}(x,y)) \rightarrow 1|(X^{'}(x,y), C^{'}(x,y)) \overset{U}{\leftarrow} T_X]\\
&\quad -Pr[\mathfrak{B}(X^{'}(x,y), C^{'}(x,y)) \rightarrow 0|(X^{'}(x,y), C^{'}(x,y)) \overset{U}{\leftarrow} T_X]|\\
&= |Pr[\mathfrak{B}(X^{'}(x,y), C^{'}(x,y)) \rightarrow 1|(X^{'}(x,y), C^{'}(x,y)) \overset{U}{\leftarrow} T_X]\\
&\quad -(1 - Pr[\mathfrak{B}(X^{'}(x,y), C^{'}(x,y)) \rightarrow 1|(X^{'}(x,y), C^{'}(x,y)) \overset{U}{\leftarrow} T_X])|\\
&= |2Pr[\mathfrak{B}(X^{'}(x,y), C^{'}(x,y)) \rightarrow 1|(X^{'}(x,y), C^{'}(x,y)) \overset{U}{\leftarrow} T_X] - 1|\\
&= 2|Pr[\mathfrak{B}(X^{'}(x,y), C^{'}(x,y)) \rightarrow 1|(X^{'}(x,y), C^{'}(x,y)) \overset{U}{\leftarrow} T_X] - 1/2|.
\end{aligned} \tag{24}$$

If the sample is picked from the set $U_X - T_X$, then the map

$$C^{'}(x,y) \mapsto m_b(t) + \ell \cdot C^{'}(x,y)(= c(x,y))(\in \mathfrak{F}_{\Gamma_e}/R_q)$$

is invertible, since

$$c(x,y) \mapsto \ell^{-1}(c(x,y) - m_b(t))(\in \mathfrak{F}_{\Gamma_e}/R_q).$$

Then, $c(x,y)$ is uniformly randomly in $\mathfrak{F}_{\Gamma_e}/R_q$, and independent of b. It follows that $\mathfrak{B}$ outputs 1 with probability 1/2.

We are able to compute $Adv^{\text{IE-LWE}}_{\mathfrak{B}}(k)$ as follows.

$$\begin{aligned}
Adv^{\text{IE-LWE}}_{\mathfrak{B}}(k) &= |Pr[\mathfrak{B}(X^{'}(x,y), C^{'}(x,y)) \rightarrow 1|(X^{'}(x,y), C^{'}(x,y)) \overset{U}{\leftarrow} T_X]\\
&\quad -Pr[\mathfrak{B}(X^{'}(x,y), C^{'}(x,y)) \rightarrow 1|(X^{'}(x,y), C^{'}(x,y)) \overset{U}{\leftarrow} U_X - T_X]|\\
&= |Pr[\mathfrak{B}(X^{'}(x,y), C^{'}(x,y)) \rightarrow 1|(X^{'}(x,y), C^{'}(x,y)) \overset{U}{\leftarrow} T_X] - 1/2|
\end{aligned}$$

Comparing the equation (24), we have

$$Adv_{\Sigma,\mathfrak{A}}^{\text{IND-CPA}}(k) = 2 \cdot Adv_{\mathfrak{B}}^{\text{IE-LWE}}(k).$$

This is a contradiction to the assumption, since a polynomial time algorithm $\mathfrak{B}$ satisfying $Adv_{\mathfrak{B}}^{\text{IE-LWE}}(k) \geq \epsilon(k)/2$ can be constructed. We conclude the desired claim.

In addition, one can make the Giophantus encryption scheme IND-CCA2 secure by using well-known conversions, such as those in [20]. However, the converted scheme is no longer homomorphic.

7 Security analysis

In this section, we introduce two possible attacks for the IE-LWE assumption. However, other attacks against ASC, which this scheme was developed from, cannot be applied to this problem. For example, the ideal decomposition attack described in section 4.1 does not work on our scheme because our scheme does not have a product structure such as $m(x,y)s(x,y)$ in (14).

From this section, we assume $\deg X(x,y) = \deg r(x,y) = 1$ and $\ell = 4$.

7.1 The Linear Algebra attack

For a given pair of polynomials $(X(x,y), Y(x,y))$, we can determine that $(X(x,y), Y(x,y))$ is sampled from T_X if we find $r \in \mathfrak{F}_{\Gamma_r}/R_q$ and $e \in \mathfrak{F}_{\Gamma_{Xr}}/R_\ell$ such that $Y(x,y) = X(x,y)r(x,y) + e(x,y)$.

The IE-LWE searching problem, which finds polynomials $r(x,y)$ and $e(x,y)$ of this type, can be solved by using the Linear Algebra attack (see Section 4.1) as follows. We construct a system of linear equations by comparing the coefficients of $x^i y^j$ in the relation

$$\sum_{(i,j)\in\Gamma_e} d_{ij}(t)x^i y^j = \left(\sum_{(i,j)\in\Gamma_X} a_{ij}(t)x^i y^j\right)\left(\sum_{(i,j)\in\Gamma_r} r_{ij}(t)x^i y^j\right) + \left(\sum_{(i,j)\in\Gamma_e} e_{ij}(t)x^i y^j\right), \tag{25}$$

where $r_{ij}(t)$ and $e_{ij}(t)$ are elements of R_q and R_ℓ, respectively.

In the case $\deg X = \deg r = 1$, we can express X,r,e, and Y in the following manner.

$$\begin{aligned}
X(x,y) &= a_{10}(t)x + a_{01}(t)y + a_{00}(t),\\
r(x,y) &= \ r_{10}(t)x + r_{01}(t)y + r_{00}(t),\\
e(x,y) &= \ e_{20}(t)x^2 + e_{11}(t)xy + e_{02}(t)y^2 + e_{10}(t)x + e_{01}(t)y + e_{00}(t),\\
Y(x,y) &= d_{20}(t)x^2 + d_{11}(t)xy + d_{02}(t)y^2 + d_{10}(t)x + d_{01}(t)y + e_{00}(t).
\end{aligned} \tag{26}$$

In this section, we employ a small example (27),

$$\begin{array}{ll} X(x,y) &= (818+1072t)x + (301+264t)y + (371+916t), \\ (u_x,u_y) &= (1+3t, 3+2t), \\ r(x,y) &= (1234+83t)x + (188+675t)y + (853+1285t), \\ e(x,y) &= 3x^2 + (2+t)xy + 3ty^2 + (1+2t)x + 2y + (2+t), \end{array} \tag{27}$$

to clarify the attack procedure. Here, $n=2$, $\ell=4$, $q=1459$, and a small solution (u_x,u_y) satisfies $X(u_x(t),u_y(t))=0$. Then, $Y(x,y)(=X(x,y)r(x,y)+e(x,y))$ is

$$\begin{array}{ll} Y(x,y) = & (1223+315t)x^2 + (1402+1442t)xy + (1348+403t)y^2 + (425+48t)x \\ & +(123+179t)y + (968+426t). \end{array}$$

When this example (X,Y) is given by the IE-LWE oracle, we can establish simultaneous linear equations (28) by comparing coefficients from both sides of the equation $Y(x,y) = X(x,y)r(x,y)+e(x,y)$, where $r(x,y)$ and $e(x,y)$ are unknown.

$$\begin{array}{l} a_{10}(t)r_{10}(t) + e_{20}(t) = d_{20}(t), \\ a_{10}(t)r_{01}(t) + a_{01}(t)r_{10}(t) + e_{11}(t) = d_{11}(t), \\ a_{01}(t)r_{01}(t) + e_{02}(t) = d_{02}(t), \\ a_{10}(t)r_{00}(t) + a_{00}(t)r_{10}(t) + e_{10}(t) = d_{10}(t), \\ a_{01}(t)r_{00}(t) + a_{00}(t)r_{01}(t) + e_{01}(t) = d_{01}(t), \\ a_{00}(t)r_{00}(t) + e_{00}(t) = d_{00}(t). \end{array} \tag{28}$$

In the case of example (27), we can write $r_{ij}(t) = r_{ij0} + r_{ij1}t$, where r_{ij0} and r_{ij1} are variables valued at $\{0, \cdots q-1\}$ in F_q, and also write $e_{ij}(t) = e_{ij0} + e_{ij1}t$, where e_{ij0} and e_{ij1} are variables valued at $\{0, \cdots \ell-1\}$ in F_q.

By using the example (27) and considering (X,Y), we can specify the equation (28) as follows.

$$\begin{array}{l} (818+1072t)(r_{100}+r_{101}t) + e_{200} + e_{201}t = 1223+315t, \\ (818+1072t)(r_{010}+r_{011}t) + (301+264t)(r_{100}+r_{101}t) + e_{110} + e_{111}t = 1402+1442t, \\ (301+264t)(r_{010}+r_{011}t) + e_{020} + e_{021}t = 1348+403t, \\ (818+1072t)(r_{000}+r_{001}t) + (371+916t)(r_{100}+r_{101}t) + e_{100} + e_{101}t = 425+48t, \\ (301+264t)(r_{000}+r_{001}t) + (371+916t)(r_{010}+r_{011}t) + e_{010} + e_{011}t = 123+179t, \\ (371+916t)(r_{000}+r_{001}t) + e_{000} + e_{001}t = 968+426t. \end{array} \tag{29}$$

The system has the solution space with dimension at least 6 since there are 18 variables and 12 equations. In the case of $\deg X(x,y) = \deg r(x,y) = 1$, a linear system obtained by this attack has the solution space with dimension at least $3n$ since the system has $9n$ variables and $6n$ equations.

If we can find a solution such that the values $e_{ij}(t)$ are in R_ℓ, then we conclude that $(X(x,y), Y(x,y))$ is in T_X. We can find them exactly by an exhaustive search for the polynomial $e(x,y)$, but this attack can be avoided by increasing $\#\Gamma_e = 6n$ to

$$((\ell-1)\ell^{n-1})^{6n} > 2^k,$$

where k is a security parameter.

We employ a lattice-reduction attack to find a suitable small e_{ij}. Any element $a \in R_q$ can be written as a vector $(a_0, a_1, \cdots, a_{n-2}, a_{n-1})$ for

$$a = a_0 + a_1 t + \cdots + a_{n-2} t^{n-2} + a_{n-1} t^{n-1}$$

on F_q. When elements $b, c \in R_q$ are written in the same manner as a, we can describe $ab + c$ as

$$\begin{pmatrix} b_0 \ b_1 \cdots b_{n-2} \ b_{n-1} \end{pmatrix} \begin{pmatrix} a_0 & a_1 & \cdots & a_{n-2} & a_{n-1} \\ a_{n-1} & a_0 & \cdots & a_{n-3} & a_{n-2} \\ a_{n-2} & a_{n-1} & \cdots & a_{n-4} & a_{n-3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_1 & a_{n-1} & \cdots & a_{n-1} & a_0 \end{pmatrix} + \begin{pmatrix} c_0 \ c_1 \cdots c_{n-2} \ c_{n-1} \end{pmatrix} \tag{30}$$

on F_q.

Using this expression, the first equation of (28) is described as

$$\boldsymbol{r_{10}} A_{10} + \boldsymbol{e_{20}} = \boldsymbol{d_{20}}, \tag{31}$$

where A_{10} is expressed as

$$A_{10} = \begin{pmatrix} a_0 & a_1 & \cdots & a_{n-2} & a_{n-1} \\ a_{n-1} & a_0 & \cdots & a_{n-3} & a_{n-2} \\ a_{n-2} & a_{n-1} & \cdots & a_{n-4} & a_{n-3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_1 & a_2 & \cdots & a_{n-1} & a_0, \end{pmatrix}$$

and r_{10},e_{20}, and d_{20} are denoted by

$$\begin{aligned} \boldsymbol{r_{10}} &= \begin{pmatrix} r_{100} & r_{101} & \cdots & r_{10n-2} & r_{10n-1} \end{pmatrix}, \\ \boldsymbol{e_{20}} &= \begin{pmatrix} e_{200} & e_{201} & \cdots & e_{20n-2} & e_{20n-1} \end{pmatrix}, \\ \boldsymbol{d_{20}} &= \begin{pmatrix} d_{200} & d_{201} & \cdots & d_{20n-2} & d_{20n-1} \end{pmatrix}, \end{aligned}$$

respectively. By using our example, this relation can be described as

$$\begin{pmatrix} r_{100} \ r_{101} \end{pmatrix} \begin{pmatrix} 818 & 1072 \\ 1072 & 818 \end{pmatrix} + \begin{pmatrix} e_{200} \ e_{201} \end{pmatrix} = \begin{pmatrix} 1223 \ 315 \end{pmatrix},$$

where each element is in F_q.

To apply lattice reduction to (31), we add the integer vector

$$\boldsymbol{u_{20}} = (u_{200}, \cdots, u_{20n-1})$$

to (31), such as

$$\boldsymbol{r_{10}} A_{10} + q\boldsymbol{u_{20}} + \boldsymbol{e_{20}} = \boldsymbol{d_{20}}. \tag{32}$$

This equation is defined over the integer ring $\mathbb{Z}$. Then we can consider an integer lattice

$$\mathcal{L}_{LAA_1} = \begin{pmatrix} A_{10} \\ qI_n \end{pmatrix},$$

where I_n denotes the $n \times n$ identity matrix. By using the example (27),

$$\begin{pmatrix} r_{100} \; r_{101} \; u_{100} \; u_{101} \end{pmatrix} \begin{pmatrix} 818 & 1072 \\ 1072 & 818 \\ 1459 & 0 \\ 0 & 1459 \end{pmatrix} + \begin{pmatrix} e_{200} \; e_{201} \end{pmatrix} = \begin{pmatrix} 1223 \; 315 \end{pmatrix} .$$

If we find a point v closest to $\boldsymbol{d_{20}}$ in the lattice $\mathcal{L}_{LAA_1}$, then we can conclude that $\boldsymbol{d_{20}} - \boldsymbol{v} = \pm\boldsymbol{e_{20}}$ with high probability since

$$\boldsymbol{d_{20}} - \boldsymbol{r_{10}} A_{10} - q\boldsymbol{u_{20}} = \pm\boldsymbol{e_{20}} .$$

Therefore, we need to find the vector closest to $\boldsymbol{d_{20}}$ in the lattice $\mathcal{L}_{LAA_1}$ to find $\boldsymbol{e_{20}}$, since the vectors $\boldsymbol{r_{20}}$ and $\boldsymbol{u_{20}}$ corresponding to $\boldsymbol{e_{20}}$ are found at the same time.

In the same way, $\pm\boldsymbol{e_{11}}$,$\boldsymbol{r_{10}}$, and $\boldsymbol{r_{01}}$ can be detected from a point w closest to the $\boldsymbol{d_{11}}$ in the lattice

$$\mathcal{L}_{LAA_2} = \begin{pmatrix} A_{10} \\ A_{01} \\ qI_n \end{pmatrix} .$$

By using our example,

$$\mathcal{L}_{LAA_2} = \begin{pmatrix} 301 & 264 \\ 264 & 301 \\ 818 & 1072 \\ 1072 & 818 \\ 1459 & 0 \\ 0 & 1459 \end{pmatrix} .$$

Therefore, we need to consider all equations in (28) simultaneously. Doing so, we see that the linear algebra attack can be reduced to the closest-vector problem (CVP) on the lattice

$$\mathcal{L}_{LAA} = \begin{pmatrix} A_{10} & A_{01} & & A_{00} & & \\ & A_{10} & A_{01} & & A_{00} & \\ & & & A_{10} & A_{01} & A_{00} \\ qI_n & & & & & \\ & qI_n & & & & \\ & & qI_n & & & \\ & & & qI_n & & \\ & & & & qI_n & \\ & & & & & qI_n \end{pmatrix} \tag{33}$$

and the vector $\boldsymbol{d} = \begin{pmatrix} \boldsymbol{d_{20}} \; \boldsymbol{d_{11}} \; \boldsymbol{d_{02}} \; \boldsymbol{d_{10}} \; \boldsymbol{d_{01}} \; \boldsymbol{d_{00}} \end{pmatrix}$, where the blank spaces in (28) indicate zero matrices.

More specifically, the lattice (33) is described as follows.

$$\mathcal{L}_{LAA} = \begin{pmatrix}
818 & 1072 & 301 & 264 & 0 & 0 & 371 & 916 & 0 & 0 & 0 & 0 \\
1072 & 818 & 264 & 301 & 0 & 0 & 916 & 371 & 0 & 0 & 0 & 0 \\
0 & 0 & 818 & 1072 & 301 & 264 & 0 & 0 & 371 & 916 & 0 & 0 \\
0 & 0 & 1072 & 818 & 264 & 301 & 0 & 0 & 916 & 371 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 818 & 1072 & 301 & 264 & 371 & 916 \\
0 & 0 & 0 & 0 & 0 & 0 & 1072 & 818 & 264 & 301 & 916 & 371 \\
1459 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1459 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1459 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1459 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1459 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1459 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1459 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1459 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1459 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1459 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1459 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1459
\end{pmatrix}$$

The Hermite normal form is calculated as

$$B = \begin{pmatrix}
1 & 0 & 0 & 0 & 116 & 982 & 0 & 0 & 447 & 93 & 1220 & 712 \\
0 & 1 & 0 & 0 & 982 & 116 & 0 & 0 & 93 & 447 & 712 & 1220 \\
0 & 0 & 1 & 0 & 1257 & 183 & 0 & 0 & 239 & 1311 & 0 & 0 \\
0 & 0 & 0 & 1 & 183 & 1257 & 0 & 0 & 1311 & 239 & 0 & 0 \\
0 & 0 & 0 & 0 & 1459 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1459 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1257 & 183 & 239 & 1311 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 183 & 1257 & 1311 & 239 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1459 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1459 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1459 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1459
\end{pmatrix}.$$

This is a special case of a q-ary lattice, such as

$$\begin{pmatrix} I & A \\ O & qI \end{pmatrix}. \tag{34}$$

Here, A consists of sparse cyclic matrices.

While the CVP on lattices is NP-hard, we need to apply known approximation algorithms for solving CVP to evaluate appropriate parameters. This paper introduces the embedding technique, which is an efficient method to solve CVP. To simplify, we start by describing the embedding technique in the case of $\deg X(x,y) = \deg r(x,y) = 1$. The relation $\boldsymbol{r}A + q\boldsymbol{u} + \boldsymbol{e} = \boldsymbol{d}$ is satisfied, where

$$\begin{aligned}
\boldsymbol{r} &= \begin{pmatrix} r_{100} \; r_{101} \; r_{010} \; r_{011} \; r_{000} \; r_{001} \end{pmatrix}, \\
\boldsymbol{u} &= \begin{pmatrix} u_{200} \; u_{201} \; u_{110} \; u_{111} \; u_{020} \; u_{021} \; u_{100} \; u_{101} \; u_{010} \; u_{011} \; u_{000} \; u_{001} \end{pmatrix}, \\
\boldsymbol{e} &= \begin{pmatrix} e_{200} \; e_{201} \; e_{110} \; e_{111} \; e_{020} \; e_{021} \; e_{100} \; e_{101} \; e_{010} \; e_{011} \; e_{000} \; e_{001} \end{pmatrix}.
\end{aligned}$$

Since the vector $\boldsymbol{e}$ is short, we may find $\boldsymbol{e}$ by calculating the vector in the lattice $(A|qI_n)$ closest to the vector $\boldsymbol{d}$. If vector $\boldsymbol{c}$ is the closest vector, then there is a possibility that the vector $\boldsymbol{e}$ is equal to the vector $\pm(\boldsymbol{d}-\boldsymbol{c})$. In our example, the correct vector of $\boldsymbol{e}$ is

$$\boldsymbol{e} = \begin{pmatrix} 3\ 0\ 2\ 1\ 0\ 3\ 1\ 2\ 2\ 0\ 2\ 1 \end{pmatrix}. \tag{35}$$

This paper shows computational experiments intended to find the closest vector by the embedding technique.

The embedding technique finds the closest vector from the lattice found by adding the target vector to the original lattice, such as

$$\mathcal{L}_d = \begin{pmatrix} B & \boldsymbol{0} \\ \boldsymbol{d} & \mu \end{pmatrix},$$

where $\boldsymbol{d}$ is a target vector and μ is a small integer, such as 1 or 2. When we reduce the lattice $\mathcal{L}_d$ by applying the LLL or BKZ method, we can find the vector $\boldsymbol{e}$ as a row vector whose last element equals μ or $-\mu$ in the reduced lattice.

For the example (27), the embedded lattice is

$$\begin{pmatrix}
818 & 1072 & 301 & 264 & 0 & 0 & 371 & 916 & 0 & 0 & 0 & 0 & 0 \\
1072 & 818 & 264 & 301 & 0 & 0 & 916 & 371 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 818 & 1072 & 301 & 264 & 0 & 0 & 371 & 916 & 0 & 0 & 0 \\
0 & 0 & 1072 & 818 & 264 & 301 & 0 & 0 & 916 & 371 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 818 & 1072 & 301 & 264 & 371 & 916 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1072 & 818 & 264 & 301 & 916 & 371 & 0 \\
1459 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1459 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1459 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1459 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1459 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1459 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1459 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1459 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1459 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1459 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1459 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1459 & 0 \\
1223 & 315 & 1402 & 1442 & 1348 & 403 & 425 & 48 & 123 & 179 & 968 & 426 & 2
\end{pmatrix},$$

since the vector $\boldsymbol{d}$ is $\begin{pmatrix} 1223\ 315\ 1402\ 1442\ 1348\ 403\ 425\ 48\ 123\ 179\ 968\ 426 \end{pmatrix}$. Applying LLL to the lattice, we can detect a shortest vector

$$\begin{pmatrix} 3\ 0\ 2\ 1\ 0\ 3\ 1\ 2\ 2\ 0\ 2\ 1 \end{pmatrix}$$

as the first row from the reduced lattice

$$
\begin{pmatrix}
3 & 0 & 2 & 1 & 0 & 3 & 1 & 2 & 2 & 0 & 2 & 1 & 2 \\
14 & 7 & -4 & -11 & -2 & 1 & -13 & 0 & -6 & 15 & -10 & 6 & 0 \\
-11 & 14 & -12 & 15 & 6 & -3 & -7 & 10 & 3 & -1 & 7 & -4 & 2 \\
12 & 15 & -1 & -8 & 16 & -11 & -8 & 1 & 1 & 12 & 9 & -7 & 4 \\
7 & 14 & -11 & -4 & 1 & -2 & 0 & -13 & 15 & -6 & 6 & -10 & 0 \\
2 & 4 & -6 & 12 & 19 & -13 & 11 & -5 & 7 & -3 & -4 & 10 & 4 \\
1 & -15 & 20 & -1 & 3 & -3 & -12 & -9 & 6 & 23 & -7 & -2 & 2 \\
2 & -3 & -4 & -12 & 21 & 12 & -1 & 11 & -16 & -2 & -11 & -12 & 4 \\
3 & 11 & -2 & 10 & -1 & -1 & 1 & 11 & 10 & 5 & -17 & -24 & 4 \\
8 & 11 & -12 & 19 & -4 & -9 & 19 & -3 & -6 & 11 & 9 & -9 & 2 \\
7 & 16 & 17 & 2 & 9 & 7 & 5 & -3 & -22 & 0 & -3 & 3 & -18 \\
8 & -10 & -3 & 13 & -9 & -17 & -4 & 10 & -12 & -20 & -14 & 2 & 22 \\
-61 & 61 & 36 & -23 & -21 & 23 & 28 & -30 & -39 & 18 & -23 & 13 & 100
\end{pmatrix}.
$$

The vector equals the correct vector $\boldsymbol{e}$ in (35).

Subring restriction technique The linear algebra attack also works in a subring of $R_q[x,y]$[52]. The ring $R_q[x,y]$ has three types of subrings or quotient rings. These are $\tilde{R_q}[x,y]$, $\tilde{R_q}[x,f(t)]$, and $\tilde{R_q}[f(t),y]$, where $\tilde{R_q}$ is a sub-ring of R_q and $f(t)$ is an element of $\tilde{R_q}$. If n is a composite number written as $n = ab$ where a and b are integers, then the quotient polynomial $t^n - 1$ can be factored into $(t^a - 1)$ and $(t^{a(b-1)} + t^{a(b-2)} + \cdots + t^a + 1)$. The ring $F_q[t]/(t^a - 1)$ is a quotient ring of R_q. The paper [22] suggests that n be chosen as prime since our scheme employs the same algebra R_q as NTRU. As in the section 5.1, we assume n is prime, then we consider the effect of the attack in the subrings $R_q[x,f(t)]$ and $R_q[f(t),y]$. Moreover, it is sufficient to consider $R_q[x,f(t)]$ since both subrings have the same structure.

This section describes how this technique works on the ring $R_q[x,f(t)]$ with an example (27).

Let $(X(x,y),Y(x,y))$ be a sample from a distribution of T_X or $U_X - T_X$. Then we can detect whether the pair belongs to T_X or not by solving the equation $Y(x,y) = X(x,y)r(x,y) + e(x,y)$ for $r \in \mathfrak{F}_{\Gamma_r}/R_q$ and $e \in \mathfrak{F}_{\Gamma_{Xr}}/R_\ell$. The lattice-reduction algorithms (which have complexity exponential with respect to the dimensionality of the lattice in general) can be applied as described above. By using the subring technique, we can expect to make the reduction easier than the original problem, since that allows reducing the dimension of the lattice.

For simplicity, let $f(t) = 0$. Then the polynomials (26) can be described as follows.

$$
\begin{aligned}
X(x,0) &= a_{10}(t)x + a_{00}(t), \\
r(x,0) &= \ r_{10}(t)x + r_{00}(t), \\
e(x,0) &= \ e_{20}(t)x^2 + e_{10}(t)x + e_{00}(t), \\
Y(x,0) &= d_{20}(t)x^2 + d_{10}(t)x + e_{00}(t).
\end{aligned}
$$

Recall the example (27) becomes the following.

$$\begin{array}{rl} X(x,y) = & (818+1072t)x + (301+264t)y + (371+916t), \\ Y(x,y) = & (1223+315t)x^2 + (1402+1442t)xy + (1348+403t)y^2 \\ & +(425+48t)x + (123+179t)y + (968+426t), \\ r(x,y) = & (1234+83t)x + (188+675t)y + (853+1285t), \\ e(x,y) = & 3x^2 + (2+t)xy + 3ty^2 + (1+2t)x + 2y + (2+t), \end{array} \tag{36}$$

Then, we can find some partial solution of $r(x,y)$ and $e(x,y)$, such as $e_{20}(t)$, $e_{10}(t)$, and $e_{00}(t)$, by solving the following linear equations.

$$\begin{array}{l} a_{10}(t)r_{10}(t) + e_{20}(t) = d_{20}(t) \\ a_{10}(t)r_{00}(t) + a_{00}(t)r_{10}(t) + e_{10}(t) = d_{10}(t) \\ a_{00}(t)r_{00}(t) + e_{00}(t) = d_{00}(t) \end{array} \tag{37}$$

By using the example (36) and considering (X,Y), we can specify the equation (28) as follows.

$$\begin{array}{l} (818+1072t)(r_{100}+r_{101}t) + e_{200} + e_{201}t = 1223+315t \\ (818+1072t)(r_{000}+r_{001}t) + (371+916t)(r_{100}+r_{101}t) + e_{100} + e_{101}t = 425+48t \\ (371+916t)(r_{000}+r_{001}t) + e_{000} + e_{001}t = 968+426t \end{array} \tag{38}$$

This system has the solution space whose dimension is 4 since there are 10 variables and 6 equations. In the case of $\deg X(x,y) = \deg r(x,y) = 1$, a linear system obtained in this attack has a solution space with at least $2n$ dimensions since the system has $5n$ variables and $3n$ equations.

If we can find a solution such that elements $e_{i0}(t)$ are in R_ℓ, then we can conclude that $(X(x,y), Y(x,y))$ is in T_X with non-negligible probability $1-1/q^6$. We need an n that satisfies

$$((\ell-1)\ell^{n-1})^{3n} > 2^k$$

to avoid an exhaustive attack on the polynomial $e(x,y)$, where k is a security parameter.

We can establish the lattice $\mathcal{L}_{LAA}^{R_q[x,0]}$ instead of $\mathcal{L}_{LAA}$ to find the solution by solving the CVP.

$$\mathcal{L}_{LAA}^{R_q[x,0]} = \begin{pmatrix} A_{10} & A_{00} & \\ & A_{10} & A_{00} \\ qI_n & & \\ & qI_n & \\ & & qI_n \end{pmatrix} \tag{39}$$

and the vector $\boldsymbol{d} = \begin{pmatrix} \boldsymbol{d_{20}} & \boldsymbol{d_{10}} & \boldsymbol{d_{00}} \end{pmatrix}$, where the blank spaces in (39) indicate zero matrices.

More specifically, the lattice (39) is described as follows.

$$
\mathcal{L}_{LAA} = \begin{pmatrix}
818 & 1072 & 371 & 916 & 0 & 0 \\
1072 & 818 & 916 & 371 & 0 & 0 \\
0 & 0 & 818 & 1072 & 371 & 916 \\
0 & 0 & 1072 & 818 & 916 & 371 \\
1459 & 0 & 0 & 0 & 0 & 0 \\
0 & 1459 & 0 & 0 & 0 & 0 \\
0 & 0 & 1459 & 0 & 0 & 0 \\
0 & 0 & 0 & 1459 & 0 & 0 \\
0 & 0 & 0 & 0 & 1459 & 0 \\
0 & 0 & 0 & 0 & 0 & 1459
\end{pmatrix},
$$

The Hermite normal form is calculated as B.

$$
B = \begin{pmatrix}
1 & 0 & 0 & 0 & 1220 & 712 \\
0 & 1 & 0 & 0 & 712 & 1220 \\
0 & 0 & 1 & 0 & 239 & 1311 \\
0 & 0 & 0 & 1 & 1311 & 239 \\
0 & 0 & 0 & 0 & 1459 & 0 \\
0 & 0 & 0 & 0 & 0 & 1459
\end{pmatrix}
$$

This matrix can be described as

$$
\begin{pmatrix}
I & & A \\
 & I & B \\
 & & qI
\end{pmatrix}, \tag{40}
$$

where A and B are cyclic matrices.

By using the embedding technique, we can find a shortest vector

$$
\begin{pmatrix} 3 & 0 & 1 & 2 & 2 & 1 & 2 \end{pmatrix}
$$

as the first row from the reduced lattice

$$
\begin{pmatrix}
3 & 0 & 1 & 2 & 2 & 1 & 2 \\
2 & 3 & -2 & 8 & -4 & -3 & 0 \\
-3 & -2 & 4 & 1 & 8 & 1 & -4 \\
-6 & 6 & 2 & 3 & -3 & -2 & 4 \\
-3 & -2 & -8 & 2 & 3 & 4 & 0 \\
-5 & -3 & 5 & 7 & -6 & 2 & 2 \\
1 & -9 & 4 & -3 & -1 & -5 & 6
\end{pmatrix}.
$$

The vector equals the correct vector $\boldsymbol{e}$ in (35).

Let us assume that $f(t)$ is not zero. Since the polynomial $e(x, y)$ can be described by

$$
e(x, f(t)) = e_{20}(t)x^2 + (e_{11}(t)f(t) + e_{10}(t))x + (e_{02}(t)f(t)^2 + e_{01}(t)f(t) + e_{00}(t)),
$$

the coefficients and the degree of $f(t)$ should be small enough to detect the polynomial $e(x, f(t))$ with small coefficients. Since $MC(e)$ can be estimated from $MC(e) \leq n^2 \cdot \ell MC(f(t))^2$, we can see the case $f(t) = 0$ is most effective for detecting the polynomial $e(x, f(t))$. So for the discussion of the subring technique in Section 7.2 we consider only the ring $R_q[x, f(t)]$.

7.2 Key recovery attack

If a solution $(\tilde{u}_x(t), \tilde{u}_y(t)) \in R_q^2$ to $X(x, y) = 0$ (not necessarily the secret key), in which all coefficients are less than ℓ is found, then the IE-LWE problem can be solved with high probability, as follows. For an IE-LWE instance $(X(x, y), Y(x, y))$, if all coefficients of $\ell \cdot Y(\tilde{u}_x(t), \tilde{u}_y(t))$ are multiples of ℓ, then it can be concluded that (X, Y) is sampled from T_X. In fact, sampling (X, Y) from T_X implies that

$$\begin{aligned}\ell \cdot Y(\tilde{u}_x(t), \tilde{u}_y(t)) &= \ell(X(\tilde{u}_x(t), \tilde{u}_y(t))r(\tilde{u}_x(t), \tilde{u}_y(t)) + e(\tilde{u}_x(t), \tilde{u}_y(t)) \\ &= \ell \cdot e(\tilde{u}_x(t), \tilde{u}_y(t)),\end{aligned}$$

and $MC(e(\tilde{u}_x(t), \tilde{u}_y(t))) < q$ implies that all coefficients of $\ell \cdot e(\tilde{u}_x(t), \tilde{u}_y(t))$ are multiples of ℓ. On the other hand, if $(X(x, y), Y(x, y))$ is sampled from U_X, then the probability that all coefficients of $\ell \cdot Y(\tilde{u}_x(t), \tilde{u}_y(t))$ are multiples of ℓ is about $1/\ell^n$. Therefore if a small solution, such as $(\tilde{u}_x(t), \tilde{u}_y(t))$, can be found, then the IE-LWE problem can be solved with a probability higher than $1 - 1/\ell^n$ by checking whether all coefficients of $\ell \cdot Y(\tilde{u}_x(t), \tilde{u}_y(t))$ are multiples of ℓ. Since $n, \ell \geq 2$, the probability $1 - 1/\ell^n$ is at least $3/4$, which is non-negligible.

In the following, we consider the key recovery attack on our encryption scheme (i.e., finding the small solution belonging to R_ℓ^2, to $X(x, y) = 0$ over R_q, by using lattice reduction techniques). First, we consider the case $\deg X = 1$. In this case, we need to find $u_x(t), u_y(t) \in R_\ell^2$ satisfying

$$a_{10}(t)u_x(t) + a_{01}(t)u_y(t) + a_{00}(t) = 0. \tag{41}$$

We write this equation with a matrix and vectors, in the same manner as the algebraic attack described above, as

$$\begin{pmatrix} \boldsymbol{u_x} & \boldsymbol{u_y} \end{pmatrix} \begin{pmatrix} A_{10} \\ A_{01} \end{pmatrix} = \begin{pmatrix} -\boldsymbol{a_{00}} \end{pmatrix}, \tag{42}$$

where the vectors $\boldsymbol{u_x}$ and $\boldsymbol{u_y}$ corresponding to $u_x(t)$, and $u_y(t)$, respectively, with elements restricted to $\{0, \cdots, \ell - 1\}$ in F_q.

As the Linear Algebra attack, we apply lattice reduction to find a small solution $(u_x(t), u_y(t))$. Then we add the integer vector $\boldsymbol{u}$

$$\boldsymbol{u} = (u_0, \cdots, u_{n-1})$$

to (41), such as

$$\boldsymbol{u_x}A_{10} + \boldsymbol{u_y}A_{01} + q\boldsymbol{u} = -\boldsymbol{a_{00}}. \tag{43}$$

This equation is defined over the integer ring $\mathbb{Z}$. We consider an integer matrix

$$A = \begin{pmatrix} A_{10} \\ A_{01} \\ qI_n \end{pmatrix}$$

and

$$\begin{pmatrix} \boldsymbol{u_x} \ \boldsymbol{u_y} \ \boldsymbol{u} \end{pmatrix} \begin{pmatrix} A_{10} \\ A_{01} \\ qI_n \end{pmatrix} = \begin{pmatrix} -\boldsymbol{a_{00}} \end{pmatrix}. \tag{44}$$

Then, we consider the lattice

$$\mathcal{L}_{KRA} = \{ \ \boldsymbol{x} \in R_q^3 \mid \boldsymbol{x}A = \boldsymbol{0} \ \} \tag{45}$$

and let $\boldsymbol{v}$ be a solution to the system (41). Then, any solution of (41) can be written as $\boldsymbol{v} + \boldsymbol{w}$ ($\boldsymbol{w} \in \mathcal{L}_{KRA}$). Observe that our target solution $(\boldsymbol{u_x}, \boldsymbol{u_y}, \boldsymbol{u})$ of (41) is expected to be relatively short among the solutions of (41) because all of the coefficients of $u_x(t)$ and $u_y(t)$ are restricted to $\{0, \cdots, \ell - 1\}$, where ℓ is much smaller than q. This observation leads us to an approach to the key-recovery attack as follows. First, we solve the system (41) and find its solution space $\mathcal{L}_{KRA}$ and a solution $\boldsymbol{v}$. Second, we solve CVP to find the vector $\boldsymbol{w}$ closest to $\boldsymbol{v}$, and then $\boldsymbol{v} - \boldsymbol{w}$ is the smallest solution of (41) and is expected to be our target solution $(\boldsymbol{u_x}, \boldsymbol{u_y}, \boldsymbol{u})$.

We provide an example (46) to see the relation in a concrete manner.

$$\begin{aligned} X(x, y) = &\ (968 + 302t)x + (861 + 442t)y + (1109 + 271t) \\ (u_x, u_y) = &\ (2 + 2t, 1 + 3t) \end{aligned} \tag{46}$$

Here, $n = 2$,$\ell = 4$, and $q = 1459$. Then we define a small solution (u_x, u_y) as

$$\begin{aligned} u_x &= u_{x0} + u_{x1}t, \\ u_y &= u_{y0} + u_{y1}t, \end{aligned}$$

where u_{x0},u_{x1},u_{y0}, and u_{y1} are variables valued at $\{0, \cdots, \ell - 1\}$ in F_q. This gives

$$(968 + 302t)(u_{x0} + u_{x1}t) + (861 + 442t)(u_{y0} + u_{y1}t) + (1109 + 271t) = 0.$$

Moreover, we can transfer the above formula to the polynomial ring $\mathbb{Z}[t]$ by adding $q \cdot u$, which is described by $u = u_0 + u_1 t$. Thus,

$$(968{+}302t)(u_{x0}{+}u_{x1}t){+}(861{+}442t)(u_{y0}{+}u_{y1}t){+}1459(u_0{+}u_1t){+}(1109{+}271t) = 0.$$

We can describe the above formula as

$$A = \begin{pmatrix} 968 & 302 \\ 302 & 968 \\ 861 & 442 \\ 442 & 861 \\ 1459 & 0 \\ 0 & 1459 \end{pmatrix}$$

and

$$\mathcal{L}_{KRA} = \begin{pmatrix} 1 & 0 & 1014 & 1033 & -912 & -917 \\ 0 & 1 & 1033 & 1014 & -917 & -912 \\ 0 & 0 & 1459 & 0 & -861 & -442 \\ 0 & 0 & 0 & 1459 & -442 & -861 \end{pmatrix}, \tag{47}$$

which is the same as the Hermite normal form. We can describe the lattice (47) as

$$\mathcal{L}_{KRA} = \begin{pmatrix} I_n & A & C \\ O & qI_n & D \end{pmatrix}, \tag{48}$$

where A, C, and D are cyclic matrices.

We also apply the embedding technique to find the lattice point of $\mathcal{L}_{KRA}^{+}$ that is closest to the solution $\boldsymbol{v}$. Let

$$\mathcal{L}_{KRA}^{+} = \begin{pmatrix} B & \mathbf{0}^T \\ \boldsymbol{v} & \mu \end{pmatrix},$$

where $\mu = 2$ and B is the lattice $\mathcal{L}_{KRA}$, and $\boldsymbol{v}$ is a vector whose dimension is n.

Applying lattice reduction to the lattice $\mathcal{L}_{KRA}^{+}$, we expect to find the vector $(u_x(t), u_y(t), u(t), \pm\mu)$ as the row vector whose last element is equal to μ or $-\mu$.

For the example, since we can take a solution to the system (41)

$$\boldsymbol{v} = \begin{pmatrix} 261060 & 0 & 0 & -458 & -173067 & -53767 \end{pmatrix},$$

we construct an embedding lattice such that

$$\mathcal{L}_{KRA}^{+} = \begin{pmatrix} 1 & 0 & 1014 & 1033 & -912 & -917 & 0 \\ 0 & 1 & 1033 & 1014 & -917 & -912 & 0 \\ 0 & 0 & 1459 & 0 & -861 & -442 & 0 \\ 0 & 0 & 0 & 1459 & -442 & -861 & 0 \\ 261060 & 0 & 0 & -458 & -173067 & -53767 & 2 \end{pmatrix},$$

and we obtained a reduced lattice

$$\begin{pmatrix} 2 & 2 & 1 & 3 & -4 & -4 & 2 \\ 7 & 7 & -16 & -4 & 0 & 0 & 12 \\ 1 & 3 & 20 & -16 & -9 & 1 & 2 \\ -13 & -12 & -1 & 6 & 0 & 5 & 26 \\ 35 & -42 & 6 & 4 & -17 & 17 & -6 \end{pmatrix}$$

by applying the LLL algorithm. So we find the shortest vector

$$v = \begin{pmatrix} 2 & 2 & 1 & 3 & -4 & -4 & 2 \end{pmatrix},$$

which corresponds to $(u_{x0}, u_{x1}, u_{y0}, u_{y1}, u_0, u_1, \mu)$ from the first row. The vector $\begin{pmatrix} 2 & 2 & 1 & 3 \end{pmatrix}$ is equal to the correct vector $(u_{x0}, u_{x1}, u_{y0}, u_{y1}) = (2, 2, 1, 3)$.

Recent lattice attacks, such as the lattice-decoding attack and the subfield-lattice attack, do not apply to our scheme. See Subsection 7.3 for details.

Kernel technique Moreover, we applied the same reduction to the lattice

$$\mathcal{L}'_{KRA} = \begin{pmatrix} I & A \\ O & qI \end{pmatrix}, \tag{49}$$

omitting the cyclic matrices C and D from the original $\mathcal{L}_{KRA}$ since these matrices are not related to $\boldsymbol{u_x}$ and $\boldsymbol{u_y}$ directly. If we can get a small solution from $\mathcal{L}'_{KRA}$, then we should consider that reduction.

The result in table 5 shows that the attack is also valid, that is, small solutions can be found by the reduction. The result tells us this attack is the most effective against the proposed cryptosystem since the attack did not fail until $n \leq 120$, which is larger than $\mathcal{L}_{KRA}$.

Dominant attack to the proposed system By the discussion of the last two sections, dimension of the lattice to attack as follows.

Attack	Original LLA	Improved LLA	KRA
Rank	$6n$	$3n$	$2n$

This table shows the key recovery attack is dominant to evaluate security of the proposed system.

7.3 Further discussion on lattice attacks

In this section, we discuss and analyze the impact of other lattice attacks, such as a subfield lattice attack [28] and the evaluate at one attack [51], on the proposed primitive.

Subfield lattice attack Here, we discuss the subfield lattice attack in the context of our scheme. This attack can be applied to homomorphic variants of NTRU. The attack reduces the lattice problem on certain number fields to the problem on their appropriate subfields by using norm maps from the original number fields to the subfields.

NTRU variants (i.e., the NTRU on $\mathbb{Z}_q[x]/(x^{2^k}+1)$ and $\mathbb{Z}_q[x]/(x^p - x - 1)$ with prime numbers p and positive integers q) have been addressed in previous experiments by Kirchner et al. [28, Section 5]. There is no proper nontrivial subfield of the number field $\mathbb{Q}[x]/(x^p - x - 1)$, but the attack on $\mathbb{Z}_q[x]/(x^p - x - 1)$ succeeds for many parameters. We infer that the size of the parameter q is strongly related to the success or failure of the attack. As the size of q increases, the volume of the lattice becomes larger, and SVP on the lattice becomes easier. In fact, the subfield attacks on NTRU with relatively small q fail in some cases (see [28, Figures 1 and 2]). Moreover, the form $h = f/g$ of the public key for NTRU seems to have a positive effect on the attack, where f and g are secret polynomials with small coefficients and f is invertible in $\mathbb{Z}_q[x] = (x^{2^k}+1)$ or $\mathbb{Z}_q[x] = (x^p - x - 1)$.

However, when comparing Table 9 in this paper with [28, Figures 1 and 2], it is evident that the size of q in our scheme is much smaller than that in the NTRU variants. Moreover, there is a gap between the forms of the keys (public/secret keys) in our scheme and those in the above NTRU variants. The data show that the lattices derived from the two attacks on our scheme are very different from those derived from the subfield attacks on the above NTRU variants. Therefore, the subfield attack does not appear to be applicable to our scheme.

Evaluating at one attack We discuss the attack which breaks some scheme on R_q in the sense of IND-CPA by evaluating the cipher at $t = 1$. In our scheme, the attack works as follows [51]:

1. Evaluate $X(x, y, t) = 0$ on R_q at $t = 1$, namely consider the equation $X(x, y, 1) = 0$ on F_q.
2. Find a small solution (s_x, s_y) of $X(x, y, 1) = 0$ on F_q by the exhaustive attack. We have a solution such that
$$(s_x, s_y) = (u_x(1), u_y(1)) \tag{50}$$
which derived from the secret key by the natural homomorphism $R_q \to F_q$. When we denote the secret key by
$$(u_x(t), u_y(t)) = (\sum_{i=0}^{n} a_i t^i, \sum_{i=0}^{n} b_i t^i),$$
the size of each element of the solution (s_x, s_y) is upper-bounded by
$$0 \le s_x, s_y \le \max(\sum_{i=0}^{n} a_i, \sum_{i=0}^{n} b_i) \le n(\ell - 1)\ . \tag{51}$$
Then we can find a small solution (s_x, s_y) by substituting total $n^2(\ell - 1)^2$ candidates or by solving univariate equation at most $n(\ell - 1)$ times since $n(\ell - 1)$ is much smaller than q. In the following experiment, we found that the number of small solutions is only one in almost all cases.
3. Calculate
$$c(s_x, s_y, 1) \equiv m_b(1) + \ell \cdot e(s_x, s_y, 1) \mod q, \tag{52}$$
where $c(x, y, t)$ is a cipher text which is encrypted $m_b(t)$ as $c(x, y, t) = m_b(t) + X(x, y, t)r(x, y, t) + \ell \cdot e(x, y, t)$, where $b = 0$ or 1.
4. Find
$$m_b(1) + \ell \cdot e(s_x, s_y, 1) \equiv m_b(1) \mod \ell. \tag{53}$$
5. Distinguish $b = 0$ or 1 by observing the value of $m_b(1) \mod \ell$ under the condition of $m_0(1) \not\equiv m_1(1) \mod \ell$.

However, the step 4 does not always work by the following reason. The small solution (s_x, s_y) is expected within the range of inequality (51). So the attack requires q which is larger than $\max\{c(s_x, s_y, 1) \mid 0 \le s_x, s_y \le \ell - 1\}$, where

$$\begin{aligned} c(s_x, s_y, 1) &= m(1) + \ell \cdot e(s_x, s_y) \\ &= m(1) + \ell \cdot \textstyle\sum_{(i,j)\in\Gamma_e} e_{ij}(1) s_x^i s_y^j \end{aligned} \tag{54}$$

By the same reason of the inequality (51), the relation

$$0 \le m(1), e_{ij}(1) \le n(\ell - 1) \tag{55}$$

holds. Then we conclude that q have to be larger than

$$\begin{aligned} \max\{c(s_x, s_y, 1) \mid 0 \le s_x, s_y \le \ell - 1\} &= n(\ell-1) + \ell \cdot \textstyle\sum_{(i,j)\in\Gamma_e} (n(\ell-1))^{i+j+1} \\ &= n(\ell-1) + \ell \cdot \textstyle\sum_{k=0}^{dX+dr} {}_2H_k (n(\ell-1))^{k+1} \\ &= n(\ell-1) + \ell \cdot \textstyle\sum_{k=0}^{dX+dr} (k+1)(n(\ell-1))^{k+1} \end{aligned}\ , \tag{56}$$

which is n times larger than the modulus q used in our scheme (see inequality (15)).

We carried out two experiments to see the practical distribution of $c(s_x, s_y, 1) \mod \ell$. First one employs samples of legitimate cipher polynomial $c(x, y, t)$ with fixed public-key $X(x, y)$. These samples are encrypted plaintext $m(t)$ chosen from R_q uniformly at random under the condition of $m(1) \equiv 1 \mod \ell$. We call this normal case. Second one employs samples of random polynomial which are chosen from $\mathfrak{F}_{\Gamma_{Xr}}/R_q$ uniformly at random, where $\mathfrak{F}_{\Gamma_{Xr}}/R_q$ is a polynomial set which is defined in (11) and (12). Hence the elements of $\mathfrak{F}_{\Gamma_{Xr}}/R_q$ are polynomials having the same terms as the legitimate cipher polynomial $c(x, y, t)$. We call this random case.

Beullens et al. suggest an indicator "Distinguishing advantage" to see the bias of the distribution $c(s_x, s_y, 1) \mod \ell$ [14]. The distinguishing advantage is defined as the value of the probability of the 2 most likely values minus the probability of the 2 least likely values. They pointed out the distinguishablity in the parameter settings of $n = 1201, 1733, 2267$ and the q equals to $2^{31} - 1 (= 2147483647)$, where these settings are only used in KAT of the proposal [5] since $q = 2^{31} - 1$ allows efficient reduction.

We conducted an experiment to follow their suggestion with 100,000 samples each. Table 2 describes the result. In this setting, we can recognize the difference of distinguishing advantages between normal case and random case. Therefore in the setting of $n = 1201, 1733, 2267$ and the fixed q equals to $2^{31} - 1$, we conclude that our scheme is not secure in the sense of IND-CPA as Beullens et al. pointed out.

Table 2. Distribution of the experimental results due to the evaluating at one attack for the fixed $q = 2^{31} - 1$

n	q	number of samples in $c(s_x, s_y, 1) \mod \ell$									
		normal					random				
		0	1	2	3	Dist. adv.	0	1	2	3	Dist. adv.
1201	$2^{31}-1$	703	1167	52688	45442	0.9626	24844	24900	25255	25001	0.00512
1733	$2^{31}-1$	36852	28222	13412	21514	0.30148	25038	24946	24983	25033	0.00142
2267	$2^{31}-1$	24747	25522	25218	24513	0.0148	25094	25056	25120	24730	0.00428

On the other hand, we will define the appropriate parameters in Section 8 to be satisfied the following three properties.

1. They set q as the minimum prime satisfied the condition (15) which is necessary and sufficient to decrypt any ciphertext.
2. They are resistant against the linear algebra attack and the key recovery attack described in Section 7.1 and 7.2 respectively.
3. They are corresponding to NIST category I, III and V [40].

The $q = 2^{31} - 1$ is not employed in the appropriate parameters since it is not minimum prime which satisfy the condition (15). (See Section 8)

We conducted an experiment to see the distribution of normal case and random case with n in the range of 100 to 1000 and corresponding q which is satisfied the first condition of appropriate parameters. Table 3 describes the result of the experiment with 10,000 samples each. Table 3 tells us the distinguishing advantage of normal case is much more than that of random case at small n though, they become close to each other as the n increases.

Table 3. Distribution of the experimental results due to the evaluating at one attack for n fron 100 to 1000 with 10,000 samples

n	q	number of samples in $c(s_x, s_y, 1) \mod \ell$									
		normal					random				
		0	1	2	3	Dist. adv.	0	1	2	3	Dist. adv.
100	3247243	350	4563	4712	375	0.855	2518	2525	2489	2468	0.0086
200	12974419	4138	4161	875	826	0.6598	2446	2528	2536	2490	0.0128
300	29181617	3990	3036	1125	1849	0.4052	2435	2513	2521	2531	0.0104
400	51868867	2375	3155	2709	1761	0.1728	2503	2529	2600	2368	0.0258
500	81036019	2655	1748	2411	3186	0.1682	2535	2490	2468	2507	0.0084
600	116683243	2305	2933	2723	2039	0.1312	2479	2502	2501	2518	0.004
700	158810429	3041	2503	1958	2498	0.1088	2496	2456	2490	2558	0.0108
800	207417619	2337	2118	2677	2868	0.109	2453	2459	2567	2521	0.0176
900	262504829	2432	2377	2603	2588	0.0382	2498	2509	2490	2503	0.0024
1000	324072059	2451	2666	2560	2323	0.0452	2469	2505	2550	2476	0.011

To see the distinguishability between normal case and random case for the appropriate parameter, we carried out same experiments with 100,000 samples. The result is described in Table 4. That tells us the distributions are indistinguishable in the cases of $n = 1773, 2267$ since each distinguishing advantage of normal case is almost same as that of random case. On the other hand, since these results are strongly depending on the public-key $X(x, y)$ we should have some criteria to avoid weak public-keys including $n = 1201$ case.

Table 4. Distribution of the experimental results due to the evaluating at one attack in the appropriate parameters with 100,000 samples

n	q	number of samples in $c(s_x, s_y, 1) \mod \ell$									
		normal					random				
		0	1	2	3	Dist. adv.	0	1	2	3	Dist. adv.
1201	467424413	24769	25113	25559	24559	0.01344	24873	24922	25144	25061	0.0041
1733	973190461	25136	25035	25008	24821	0.00342	24883	24945	25032	25140	0.00344
2267	1665292879	25117	24791	25021	25071	0.00376	25121	25114	24970	24795	0.0047

8 Appropriate parameter values

This section intends to define an appropriate parameters which have three properties as follows :

1. They set q as the minimum prime satisfied the condition (15) which is necessary and sufficient to decrypt any ciphertext.
2. They are resistant against the linear algebra attack and the key recovery attack described in Section 7.1 and 7.2 respectively.
3. They are corresponding to NIST category I, III and V [40].

under the condition of $\deg X = \deg r = 1$ and $\ell = 4$.

To satisfy the above 2nd and 3rd properties, we have to clear the mathematical structure of lattice $\mathcal{L}'_{KRA}$ which is associated with the key recovery attack which is dominant under the condition (See Section 7.2).

Recall the discussion in the subsection 7.2, the lattice $\mathcal{L}'_{KRA}$ can be described as follows.

$$\mathcal{L}'_{KRA} = \begin{pmatrix} I & A \\ O & qI \end{pmatrix}, \tag{57}$$

where A is an $n \times n$ cyclic lattice and I is $n \times n$ identity matrix. Then the lattice $\mathcal{L}'_{KRA}$ is a q-ary lattice, where q is the minimum prime within the condition (15) such as

$$q > \ell - 1 + \ell \sum_{k=0}^{2} (k+1) n^k (\ell-1)^{k+1} = 3 + 4(3 + 2\cdot 3^2 n + 3\cdot 3^3 n^2) = 324n^2 + 72n + 15 \tag{58}$$

then we choose q as the smallest prime larger than $324n^2 + 72n + 15$, so we conclude q is $O(n^2)$.

8.1 Embedding technique in $\mathcal{L}'_{KRA}$

In this subsection, we investigate the structure of the lattice $\mathcal{L}^+_{KRA}$ which is described as

$$\mathcal{L}^+_{KRA} = \begin{pmatrix} I & A & 0 \\ O & qI & 0 \\ & \boldsymbol{v} & \mu \end{pmatrix} \quad , \tag{59}$$

where $\boldsymbol{v}$ is a solution to the system (41) whose dimension is n, and μ is the embedding factor. We choose $\mu = 2$. In our experiments, we used the embedding technique, which is a standard algorithm for solving CVP approximately and we use LLL/BKZ algorithm as a lattice-basis reduction algorithm.

Lattice reduction of $\mathcal{L}^+_{KRA}$ by LLL It is clear that the rank of $\mathcal{L}^+_{KRA}$ is $2n+1$ and the determinant is $2q^n$. The lattice $\mathcal{L}^+_{KRA}$ has a shortest vector that is corresponding to the smallest solution $(u_x(t), u_y(t))$ whose coefficients are restricted within $\{0, 1, 2, 3\}$. Then the average norms of the shortest vectors v as follows.

$$||v|| \sim \sqrt{7n}. \tag{60}$$

We conducted an experiment on the key-recovery attack to clear the characteristics of the lattice $\mathcal{L}'_{KRA}$ (see Section 7.2). We suppose that the key-recovery attack succeeds even if we find two polynomials with small coefficients $< \ell$ that differ from the correct secret key $(u_x(t), u_y(t))$.

Our computing environment is as follows:

- Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz
- Memory: 128 GB
- OS: Ubuntu 16.04.3
- Software: Magma V2.23-4

The experimental results given in Table 5 show that the key-recovery attack for $\deg X = 1$ failed for $n \geq 140$, which is a much higher threshold than for the linear algebra attack. Here, the experiment took LLL algorithm to reduce lattices.

In Table 5, Norm1($\mathcal{L}^+_{KRA}$) and Norm2($\mathcal{L}^+_{KRA}$) are the norms of the shortest vector v_1 and the second shortest vector v_2 in the LLL-reduced basis of lattice $\mathcal{L}^+_{KRA}$, respectively. Norm1($\mathcal{L}'_{KRA}$) is the norms of the shortest vector in the LLL-reduced basis of $\mathcal{L}'_{KRA}$. "Gap" indicates the gap of Norm1 and Norm2

Table 5. Experimental results for the key-recovery attack by using the lattice $\mathcal{L}^+_{KRA}$

n	q	Rank	Norm1 $\mathcal{L}^+_{KRA}$	Norm2 $\mathcal{L}^+_{KRA}$	Norm1 $\mathcal{L}^{'}_{KRA}$	Gap	Results	Time (s)
10	33149	21	9.54	169.25	187.24	17.74	Success	0
20	131059	41	10.39	631.88	873.65	60.80	Success	0.03
30	293791	61	14.28	1494.55	1614.63	104.64	Success	0.14
40	521299	81	14.56	2780.67	3516.06	190.98	Success	0.37
50	813623	101	19.29	6540.66	6550.24	339.12	Success	0.91
60	1170751	121	19.90	9949.59	13564.71	499.99	Success	2.02
70	1592659	141	23.09	23512.91	21762.52	1018.46	Success	3.91
80	2079401	161	23.15	36262.28	42922.11	1566.29	Success	6.89
90	2630917	181	24.72	67789.24	69317.22	2742.46	Success	11.37
100	3247243	201	27.18	100620.90	106281.33	3701.40	Success	290.89
110	3928361	221	29.26	146817.01	162823.37	5018.10	Success	417.84
120	4674289	241	29.95	276000.64	354445.93	9215.39	Success	867.92
130	5484979	261	29.44	550967.12	552815.34	18711.82	Success	1143.12
140	6360503	281	896427.36	1024128.16	864691.59	1.14	Failure	1823.41
150	7300819	301	1428000.37	1583305.28	1241301.69	1.11	Failure	2482.02
160	8305951	321	2720617.08	2822272.03	2301369.44	1.04	Failure	3271.97
170	9375871	341	4471969.71	4478011.90	4213869.15	1.00	Failure	4031.23
180	10510579	361	6013126.09	7153373.77	8735041.99	1.19	Failure	5077.99

such that Gap = Norm2/Norm1. We conclude SVP of $\mathcal{L}^+_{KRA}$ is unique-SVP since the gap increases until the attack failed. We also note that Norm2($\mathcal{L}^+_{KRA}$) is as same as Norm1($\mathcal{L}^{'}_{KRA}$) from our experiment in Table 5.

On the other hand, we may assume that

$$||\lambda_2(\mathcal{L}^+_{KRA})|| \approx GH(\mathcal{L}^{'}_{KRA}) \tag{61}$$

in [10], where $||\lambda_2(\mathcal{L}^+_{KRA})||$ denote the norm of the smallest vector linearly independent from the shortest non-zero vector in the embedding lattice $\mathcal{L}^+_{KRA}$, and $GH(\mathcal{L}^{'}_{KRA})$ is the Gaussian heuristic for the lattice $\mathcal{L}^{'}_{KRA}$, namely

$$GH(\mathcal{L}^{'}_{KRA}) = \sqrt{nq/\pi e}.$$

So we conclude the norm $||\lambda_2(\mathcal{L}^+_{KRA})||$ increases as the n increases.

An improvement of the lattice reduction Nguyen proposed an improvement of the lattice reduction to upgrade accuracy of output vector [37]. He suggests to find the shortest vector of the shifted lattice $\mathcal{L}^{\pm}_{KRA}(= \mathcal{L}^+_{KRA} + (-3/2, -3/2, \cdots, -3/2))$ instead of the lattice $\mathcal{L}^+_{KRA}$ itself. Since any component of the shortest vector of the lattice $\mathcal{L}^{\pm}_{KRA}$ is within the range of

$$\{-3/2, -1/2, 1/2, 3/2\}$$

Table 6. Experimental results for the key-recovery attack by using the lattice $\mathcal{L}_{KRA}^{\pm}$

n	q	Rank	Norm1 $\mathcal{L}_{KRA}^{\pm}$	Norm2	Norm1 $\mathcal{L}_{KRA}^{'}$	Gap	Results	Time (s)
10	33149	21	5.57	183.37	187.24	32.93	Success	0.01
20	131059	41	6.63	726.15	752.04	109.47	Success	0.03
30	293791	61	8.06	1782.44	1356.65	221.08	Success	0.12
40	521299	81	10.68	2958.64	3329.33	277.10	Success	0.36
50	813623	101	10.91	6104.53	7205.23	559.60	Success	0.92
60	1170751	121	12.17	12952.10	12542.96	1064.66	Success	1.95
70	1592659	141	13.82	18426.60	19035.19	1333.30	Success	4.63
80	2079401	161	13.86	43732.37	35248.17	3156.11	Success	7.13
90	2630917	181	14.66	69557.22	64973.58	4743.76	Success	103.7
100	3247243	201	15.87	111711.14	117019.68	7037.14	Success	264.27
110	3928361	221	15.97	182166.04	197823.16	11407.68	Success	593.77
120	4674289	241	16.31	327286.57	277991.07	20067.23	Success	814.44
130	5484979	261	18.52	477149.60	619277.31	25763.66	Success	1218.62
140	6360503	281	19.03	1072280.67	1070655.51	56357.82	Success	1883.07
150	7300819	301	19.87	1256035.05	1365477.37	63197.98	Success	2770.3
160	8305951	321	20.10	1919024.22	2301749.84	95475.02	Success	5452.6
170	9375871	341	4577367.73	5003355.38	4265512.79	1.09	Failure	5956.47
180	10510579	361	8927328.75	9479707.48	7151019.87	1.06	Failure	4939.89

the average norm of the shortest vector v' is

$$||v'|| \sim \sqrt{5n/2} \ , \tag{62}$$

which is less than the norm of the shortest vector v (see (60)). We conducted an experiment applying the above improved reduction and we describe the result in Table 6. We found that the improved attack succeeds up to $n = 160$. This is larger than the original attack which succeeds up to $n = 130$. And we can see the relation (61) holds as well. We conclude the average norm of the shortest vector should be assumed to be $\sqrt{5n/2}$ since we can see the norm of the shortest vector is similar to the theoretical value $\sqrt{5n/2}$.

Lattice reduction of $\mathcal{L}_{KRA}^{+}$ by BKZ Let $(b_1, \cdots, b_{2n+1})$ be a sufficiently reduced basis of the lattice $\mathcal{L}_{KRA}^{+}$ and we write $(b_1^*, \cdots, b_{2n+1}^*)$ as a basis which is given by Gram-Schmidt orthonormalization from the basis $(b_1, \cdots, b_{2n+1})$. We investigate the behavior of $\log ||b_i^*|| \ \ (i = 1, \cdots, 2n+1)$ at $n = 120$ which is succeeded in attacking in Table 5.

The Fig. 1 shows the behavior, where the red line and the blue line indicate the behavior in the case of $\beta = 10$ and $\beta = 20$ respectively. Here the basis $(b_1, \cdots, b_{2n+1})$ is given by BKZ-β which is implemented in Fplll 4.0.0 library. The figure tells us the lattice $\mathcal{L}_{KRA}^{+}$ satisfies the geometric series assumption (GSA) and the absolute value of the slope of the line decreases gentry as the β

Table 7. The behavior $\log_2 ||b_i^*||$ and the comparison $||b_2^*||/||b_1^*||$ and $||b_2||/||b_1||$

Beta	slope	y-intercept	correlation coefficient	p-value	$\|\|b_2^*\|\|/\|\|b_1^*\|\|$	$\|\|b_2\|\|/\|\|b_1\|\|$
10	-0.08354	32.2740	> 0.999	< 0.001	4320402	4320505
20	-0.07493	31.2279	> 0.999	< 0.001	1783504	1783497

increases. Therefore the norm of the second shortest vector decreases as the β increases.

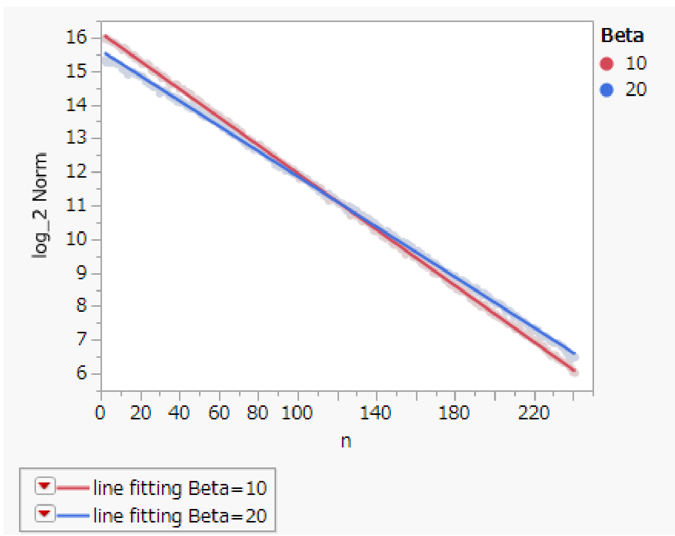

Fig. 1. Behavior of $\log_2 ||b_i^*|| \quad (i = 2, \cdots, 2n+1)$

Table 7 shows the slope and the y-intercept of the fitting line described in Fig.1 and the correlation coefficient and the p-value indicate these lines well approximate the samples. Table 7 also shows the comparison with the values of $||b_2^*||/||b_1^*||$ and $||b_2||/||b_1||$ then we observe $||b_2^*||/||b_1^*||$ is almost the same as $||b_2||/||b_1||$.

By the above observation, for the reduced basis b_i we can set as follows.

$$\begin{aligned} &||b_1^*|| = \sqrt{5n/2} \\ &||b_2^*|| = \delta^{2n}\sqrt{q} \\ &||b_i^*|| = -slope \cdot ||b_{i-1}^*|| \quad (i = 3, 4, .., 2n+1) \quad \text{with} \quad -1 < slope < 0 \ , \end{aligned} \tag{63}$$

where we assume that $||b_2^*||$ is equal to the non-zero shortest vector v obtained by the underlying lattice reduction algorithm over the lattice $\mathcal{L}'_{KRA}$ and δ denotes

Table 8. The parameters of LWE-problem and relation to our system

parameter	description	our system
n	dimension of the associated lattice	n
m	number of samples	$2n$
q	modulus	$> 324n^2 + 72n + 15$
σ	standard deviation	1.12

the root of Hermite factor which is defined

$$\delta = (||v||/(det\mathcal{L}^{'}_{KRA})^{1/2n})^{1/2n} \quad .$$

In Table 7 "slope" indicates the slope of the line of the GSA. We have the following relationship:

$$||b_1^*|| \cdot ||b_2^*|| \cdot ||b_3^*|| \cdots ||b_{2n+1}^*|| = det(\mathcal{L}^{+}_{KRA}) = 2q^n. \tag{64}$$

Then we can calculate

$$\begin{aligned} ||b_3^*|| &= \quad |slope| \cdot ||b_2^*|| \\ ||b_4^*|| &= \quad |slope|^2 \cdot ||b_2^*|| \\ &\quad \vdots \\ ||b_{2n}^*|| &= \quad |slope|^{2n-2} \cdot ||b_2^*|| \\ ||b_{2n+1}^*|| &= |slope|^{2n-1} \cdot ||b_2^*|| \quad . \end{aligned} \tag{65}$$

So we have

$$\sqrt{5n/2} \cdot |slope|^{n(2n-1)} \cdot (\delta^{2n} \cdot \sqrt{q})^{2n} = 2q^n \quad , \tag{66}$$

and

$$\log(|slope|) = (-\log(5n/2)/2 - 4n^2 \cdot \log(\delta) + \log(2))/n(2n-1) \quad . \tag{67}$$

The formula specifies the relation of $\log(|slope|)$ and δ with fixed n and q, where q is the smallest prime larger than $324n^2 + 72n + 15$ (See(58)).

8.2 Parameter estimation

In this section, we assume that the computational complexity for the key recovery attack is as same as LWE problem(n, m, q, σ) since the last subsection observes the same property of the LWE problem. The parameters n, m, q, σ are described as follows. Here $m = 2n$, and q is the smallest prime larger than $324n^2+72n+15$ in our system. The standard deviation σ of the elements of the error vector associated with LWE-problem, which is known as unique-SVP, can be calculated as $\sqrt{m}\sigma$, when we use embedding technique with the embedding factor equals 2. In our system, the average norm for elements of the shortest vector v is $\sqrt{5n/2}$ by (62). So we have $\sqrt{2n}\sigma = \sqrt{5n/2}$, then $\sigma = \sqrt{5}/2$ follows.

To estimate secure parameter n we apply "2016 Estimate" in [7], which is applied to "New Hope" [9], to our system.

First, Y.Chen suggests

$$\delta_0 = (((\pi\beta)^{1/\beta}\beta)/(2\pi e))^{1/(2(\beta-1))} \tag{68}$$

in [16], where δ_0 is the root of Hermite factor of the shortest vector obtain by BKZ with block size β. In "2016 Estimate [7]", Albrecht et al. suggest the following inequality

$$\sqrt{\beta/(2n)}\lambda_1(\mathcal{L}^+_{KRA}) \leq \delta_0^{2\beta-2n}(det\mathcal{L}^+_{KRA})^{1/2n} \ , \tag{69}$$

holds for the basis reduction by the BKZ algorithm with block size β, where $\lambda_1(\mathcal{L}^+_{KRA})$ is the norm of the non-zero shortest vector in the lattice $\mathcal{L}^+_{KRA}$, namely $\lambda_1(\mathcal{L}^+_{KRA}) \sim \sqrt{5n/2}$. So we find the pair of n and β to satisfy the both condition of (68) and (69) and estimate the complexity of lattice reduction by the formula

$$8 \cdot 2n \cdot 2^{0.292\beta+12.31} \tag{70}$$

which is given by [11].

Based on the above discussion, we design three appropriate parameters corresponding to the NIST category I, III and V as shown in Table 9, where we denote the security parameter by k. In the NIST PQC Standardization the security strength of AES128, AES192 and AES256 is estimated to be 143, 207 and 272 classical gates, respectively [40]. The security of our parameter in Table 9 was estimated by the BKZ algorithm using the subroutine cost from equation (70). However there are many different models to estimate the secure parameters of lattice-based cryptography using the BKZ algorithm [8]. Our parameters in Table 9 is slightly less than that of NIST required, namely the difference is within 13 bits. Compared with the difference appeared in the above evaluation models, the difference of 13 bits can be considered as the range of error.

Table 9. Appropriate parameters for our scheme

NIST Category	k	n	q	Secret key (bytes)	Public key (bytes)	Ciphertext (bytes)
I	135	1201	467424413	600.5	14412	28824
Ⅲ	196	1733	973190461	866.5	20796	41592
V	259	2267	1665292879	1133.5	27204	54408

9 Cryptographic scheme

This section shows a cryptographic scheme that satisfies IND-CCA2 security. This scheme is constructed by applying Fujisaki–Okamoto conversion [20] to our cryptographic primitive (which satisfies IND-CPA security as described in Section 6).

9.1 Fujisaki–Okamoto conversion

Let $\Pi := (\mathfrak{K}, \mathfrak{E}, \mathfrak{D})$ be a public-key encryption scheme that satisfies IND-CPA security, where $\mathfrak{K}$ is the key-generation algorithm, $\mathfrak{E}$ is the encryption algorithm, and $\mathfrak{D}$ is the decryption algorithm. Fujisaki–Okamoto conversion tells us that the public-key encryption scheme $\bar{\Pi} := (\bar{\mathfrak{K}}, \bar{\mathfrak{E}}^H, \bar{\mathfrak{D}}^H)$ satisfies IND-CCA2 security, such that

$$\bar{\mathfrak{E}}_{pk}^{H} = \mathfrak{E}_{pk}((x||s), H(x||s)), \tag{71}$$

where

- s is a random string chosen from an appropriate domain,
- H is a hash function
$$H : \{0,1\}^* \to \{0,1\}^{\kappa_0}$$
- $\mathfrak{E}_{pk}(message, coins)$ indicates the encryption of the indicated message using the indicated coins as random bits.

More precisely, the basic scheme $\bar{\Pi} := (\bar{\mathfrak{K}}, \bar{\mathfrak{E}}^H, \bar{\mathfrak{D}}^H)$ can be described as follows.

- Let $\bar{\mathfrak{K}}(1^\kappa) := \mathfrak{K}(1^\kappa)$.
- $\bar{\mathfrak{E}}_{pk}^H : \{0,1\}^{\kappa-\kappa_0} \times \{0,1\}^{\kappa_0} \to C$ is defined by
$$\bar{\mathfrak{E}}_{pk}^H(x, s) := \mathfrak{E}_{pk}((x||s), H(x||s)),$$
where $x \in \{0,1\}^{\kappa-\kappa_0}$, $s \in \{0,1\}^{\kappa_0}$, and C is a cipher space whose elements are valid ciphertexts.
- $\bar{\mathfrak{D}}_{sk}^H : C \to \{0,1\}^{\kappa-\kappa_0} \cup \{\perp\}$ is defined by
$$\bar{\mathfrak{D}}_{sk}^H(y) := \begin{cases} [\mathfrak{D}_{sk}(y)]^{\kappa-\kappa_0} & \text{if the condition (72) holds} \\ \perp(null) & \text{otherwise} \end{cases},$$
where $[\mathfrak{D}_{sk}(y)]^{\kappa-\kappa_0}$ indicates the first $(\kappa - \kappa_0)$ bits of $\mathfrak{D}_{sk}(y)$. The condition for ciphertext verification is
$$y = \mathfrak{E}_{pk}(\mathfrak{D}_{sk}(y), H(\mathfrak{D}_{sk}(y))). \tag{72}$$

Specifically, the following theorem holds.

Theorem 3. *Suppose Π is γ-uniform and $(t^{'}, 0, 0, \epsilon^{'})$-secure in the sense of IND-CPA. Then, for any q_H and q_D, the scheme $\bar{\Pi}$ is (t, q_H, q_D, ϵ)-secure in the sense of IND-CCA2 in the random oracle model where*

$$\begin{aligned} t &= t^{'} - q_H \cdot (T_\epsilon(\kappa) + c \cdot \kappa) \\ \epsilon &= \epsilon' \cdot (1-\gamma)^{-q_D} + q_H \cdot 2^{-(\kappa_0 - 1)}. \end{aligned} \tag{73}$$

In this, $T_\epsilon(\cdot)$ denotes the computational running time of $\mathfrak{E}_{pk}(\cdot)$ and c is a constant.

Here, γ-uniformity is defined as follows.

Definition 8. *Let $\Pi = (\mathfrak{K}, \mathfrak{E}, \mathfrak{D})$ be a public-key encryption scheme. Let the parameters mlen and clen denote the length of a plaintext message and a coins tuple s, respectively. For a given $x \in \{0,1\}^{mlen}$ and $y \in C$, define*

$$\gamma(x,y) = Pr[s \leftarrow_R \{0,1\}^{clen} : y = \mathfrak{E}_{pk}(x,s)].$$

We say that Π is γ-uniform (for any $k \in \mathbb{N}$) if, for any $x \in \{0,1\}^{mlen}$ and any $y \in C$, $\gamma(x,y) \leq \gamma$.

Now, we estimate the sizes of the parameters q_D,q_H, and γ of our encryption scheme. First, we set the parameter q_D to 2^{64}. We assume the parameter q_H is 2^{2k} to consider an exhaustive search for a quantum computer with Grover's algorithm. To calculate the parameter γ, we need to estimate the probability $\gamma(x,y)$ for any $x \in \{0,1\}^{mlen}$ and $y \in C$. Let x_0 be the fixed plaintext in $\{0,1\}^{mlen}$ and suppose the probability

$$Pr\{\forall s_1, s_2 \in \{0,1\}^{clen} | \mathfrak{E}_{pk}(x_0, s_1) = \mathfrak{E}_{pk}(x_0, s_2)\}, s_1 \neq s_2.\} \tag{74}$$

If we write

$$\mathfrak{E}_{pk}(x_0, s_i) = m(t) + X(x,y)r_i(x,y) + \ell \cdot e_i(x,y) \quad (i = 1, 2),$$

the condition (74) can be described as

$$\ell^{-1} \cdot X(x,y)(r_1(x,y) - r_2(x,y)) = e_2(x,y) - e_1(x,y).$$

If $e_1(x,y)$ equals $e_2(x,y)$, then $r_1(x,y) = r_2(x,y)$ since neither $X(x,y)$ nor ℓ equals 0. So, we can assume $e_1(x,y) \neq e_2(x,y)$. By the definition of the encryption scheme, the coefficients of $e_2(x,y) - e_1(x,y)$ are in R_q with coefficients restricted to the range $\{0, \cdots, \ell-1, q-\ell+1, \cdots, q-1\}$. Further, the coefficients of $\ell^{-1} \cdot X(x,y)(r_1(x,y) - r_2(x,y))$ are in R_q with coefficients in the range $\{0, \cdots, q-1\}$ and the coin r is in $\{0,1\}^{\kappa_0}$. Then, we can estimate the probability (74) as

$$\max(\{(2\ell/q)^{\#\Gamma_e \cdot n}, 2^{-\kappa_0}\}).$$

Since the probability does not depend on the fixed x_0, we can estimate

$$\gamma \leq \max(\{(2\ell/q)^{\#\Gamma_e \cdot n}, 2^{-\kappa_0})\}. \tag{75}$$

By the condition (15), we can estimate q as follows.

$$\begin{aligned} q &\geq \ell - 1 + \ell \textstyle\sum_{i=0}^{dX+dr}(i+1)n^i(\ell-1)^{i+1} \\ &> \ell \textstyle\sum_{i=dX+dr}^{dX+dr}(i+1)n^i(\ell-1)^{i+1} \\ &= \ell(dX+dr+1)n^{dX+dr}(\ell-1)^{dX+dr+1} \\ &> 2\ell \cdot n^2(\ell-1)^3 \\ &\geq 2\ell \cdot 2^2(\ell-1)^3 \end{aligned}$$

The last inequality is satisfied since dX and dr are each larger than or equal to 1. Then

$$(2\ell/q)^{\#\Gamma_e \cdot n} < (1/2^2(\ell-1)^3)^{\#\Gamma_e \cdot n} < 1/2^{2n}.$$

If we set ℓ equals to 4, then $1/2^{2n} = 1/2^{\kappa} < 1/2^{\kappa_0}$ is satisfied since $2n = |\ell|n = \kappa$. We conclude $\gamma < 1/2^{\kappa_0}$ in the case of $\ell = 4$.

Then, ϵ can be calculated as follows.

$$\begin{aligned}\epsilon &= \epsilon' \cdot (1 - 2^{-\kappa_0})^{-q_D} + q_H \cdot 2^{-(\kappa_0 - 1)} \\ &\sim \epsilon' \cdot (1 + (q_D + q_H) \cdot 2^{-\kappa_0})\end{aligned}$$

Since k is larger than or equal to 128,

$$\begin{aligned}\epsilon &< \epsilon' \cdot (1 + 2q_H \cdot 2^{-\kappa_0}) \\ &= \epsilon' \cdot (1 + 2 \cdot 2^{2k} \cdot 2^{-\kappa_0})\,. \\ &= \epsilon' \cdot (1 + 2^{2k+1-\kappa_0})\end{aligned}$$

According to the relation, we set $\kappa_0 \geq 2k + 1$ since ϵ is negligible (such as $\epsilon < 2\epsilon'$).

9.2 Key Generation

Same as the section 5.2.

9.3 Encryption

Recall the domain parameters as follows (See 5.1).

- ℓ: A small integer which is larger than 1.
- q: A prime which is cardinality of prime field F_q and is much larger than ℓ.
- n: Degree of the modulus polynomial of the quotient ring $R_q(= F_q[t]/(t^n - 1))$. The n should be prime for the security reason.
- dX: Total degree of the irreducible bivariate polynomial $X(x, y)$
- dr: Total degree of the random bivariate polynomial $r(x, y)$
- $mlen$ Length of the message M

1. Set the length of the payload $plen = \lceil n \cdot |\ell|/8 \rceil$
2. Create a plaintext M whose payload is $plen$ bytes in size

$$M = m||randombytes(plen - mlen)\ ,$$

   where function $randombytes(len)$ returns random data whose length is len.
3. Set the lower $8 - |\ell| \cdot (n \bmod (8/|\ell|))$ bits of M to 0.
4. Initialize the seed expander with coins equal to $H(M)$.
5. Embed a plaintext M into the coefficients of the plaintext polynomial $m(t)(\in R_\ell)$ whose degree is $n - 1$.
6. Generate a support set Γ_r of degree dr with graded lexicographic order
7. Create a random polynomial $r(x, y)$ as follows:
   (a) Set $r = 0$
   (b) For each (i, j) in Γ_r
      i. Choose a coefficient $r_{ij}(t)$ uniformly at random from the set R_q

ii. Set $r(x,y) = r(x,y) + r_{ij}(t)x^i y^j$
8. Generate a support set Γ_e of degree $dX + dr$ with graded lexicographic order
9. Create a noise polynomial $e(x,y)$ as follows:
   (a) Set $e(x,y) = 0$
   (b) For each (i,j) in Γ_e
      i. Choose a coefficient $e_{ij}(t)$ uniformly at random from the set R_ℓ
      ii. Set $e(x,y) = e(x,y) + e_{ij}(t)x^i y^j$
10. Construct the cipher polynomial $c(x,y)$ as

$$c(x,y) = m(t) + X(x,y)r(x,y) + \ell \cdot e(x,y) \tag{76}$$

9.4 Decryption

1. Substitute the secret key that is a small solution $(u_x(t), u_y(t))$ over R_q of $X(x,y) = 0$ into $c(x,y)$:

$$c(u_x(t), u_y(t)) = m(t) + \ell \cdot e(u_x(t), u_y(t)) \tag{77}$$

   When the parameters ℓ and q satisfy the relation described above (15), each coefficient of $m(t) + \ell \cdot e(u_x(t), u_y(t)) \in \mathbb{Z}/(t^n - 1)$ is within the range from 0 to $q-1$. Theorem 1 gives a proof of this fact.
2. Extract $m(t)$ from $c(u_x(t), u_y(t))$ as

$$c(u_x(t), u_y(t)) \pmod{\ell} = m(t)$$

   where we consider $c(u_x(t), u_y(t))$ as an element of $\mathbb{Z}[t]$
3. Recover the plaintext M from the coefficients of $m(t)$
4. Initialize the seed expander with coins equal to $H(M)$.
5. Encrypt the plaintext polynomial $m(t)$

$$c'(x,y) = m(t) + X(x,y)r(x,y) + \ell \cdot e(x,y)$$

6. If $c'(x,y)$ equals $c(x,y)$ then $m = [M]^{mlen}$ and $flag = valid$; otherwise, $m = null$ and $flag = invalid$

Here, $[x]^{len}$ denotes extraction of the most significant len bits of x.

10 Performance analysis

This section shows the results of the preliminary performance analysis, which is carried out by using a reference implementation and an optimized implementation (including this proposal). Table 10 and Table 11 show the cycles of each function described in Sections 9.2 to 9.4. We carried out this analysis on a platform with the following characteristics:

CPU	Xeon E5-1620 3.6GHz
OS	Windows 7, 64bit
memory	32 GB memory

Figure 2 shows the differences between these implementations.

Table 10. Performance of reference implementations

Name	NIST Category	Security (bits)	keygen (cycles)	encrypt (cycles)	decrypt (cycles)
IEC602	I	135	92909566	178456036	335353573
IEC868	Ⅲ	196	160497017	378860493	716243384
IEC1134	V	259	239510004	626677271	1186128486

Table 11. Performance of optimized implementations

Name	NIST Category	Security (bits)	keygen (cycles)	enceypt (cycles)	decrypt (cycles)
IEC602	I	135	78272627	116773401	216049724
IEC868	Ⅲ	196	131971731	248815749	466577361
IEC1134	V	259	191246205	420543208	792576864

11 Advantages

One of the advantage of the proposed cryptographic primitives described in Section 5 is that the system has homomorphic properties. Homomorphic properties allow us to compute on encrypted data without decoding. They can calculate addition and/or multiplication of single- or multiple-bit integers in the encrypted state. These properties are attractive to industries such as the smart device community and cloud computing. These industries need to handle personal data that must be kept secret from others. In the smart device community, we need to send (for example) meter data to the electric power company. Although the smart mater encrypts its output, the amount of data is too large to send at short intervals. The data must be conslidated at intermediate nodes by calculating in an encrypted state.

Table 12 describes the classification of homomorphic encryption (HE) with respect to the computable number of times and sizes of public keys and encrypted data. From the table, we see that sizes of the public keys and the ciphertext increase with increasing the computable number of times. Somewhat homomorphic encryption (SHE) and fully homomorphic encryption (FHE) are realized by lattice-based encryption such as LWE, NTRU, or Nuida–Kurosawa's scheme. It is clear that lattice-based encryption is as important as homomorphic encryption.

11.1 Homomorphic property

The Giophantus cryptographic primitives have the ring homomorphic property. When two different cipher polynomials

$$\begin{aligned} c_1(x,y) &= m_1(t) + X(x,y)r_1(x,y) + \ell \cdot e_1(x,y) \\ c_2(x,y) &= m_2(t) + X(x,y)r_2(x,y) + \ell \cdot e_2(x,y) \end{aligned} \tag{78}$$

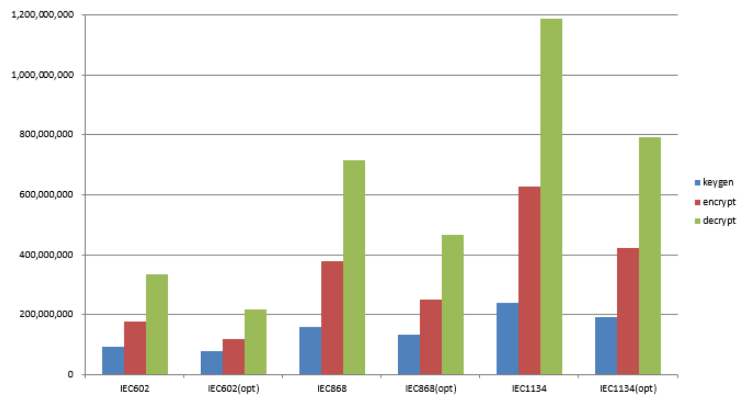

Fig. 2. Performance of the reference and optimized implementations

Table 12. Classification of homomorphic encryption

Type	Number of operations		Message	Public key	Ciphertext	Example
	Addition	Multiplication	(bit)	size	size	
HE	Any	No	Multi	Small	Small	Paillier [42]
	No	Any	Multi	Small	Small	ElGamal [18]
SHE	Any	Once	Multi	Small	Small	Pairing [12]
	Any	Several	Single	Large	Medium	LWE [31]
	Any	Several	Multi	Large	Medium	Giophantus, NTRU [48]
FHE	Any	Any	Single	Large	Large	Lattice-based [23]
	Any	Any	Multi	Large	Large	Nuida–Kurosawa [41]

are given, we define the addition and multiplication as $c_1(x,y)+c_2(x,y)$ and $c_1(x,y)c_2(x,y)$, respectively.

Additive homomorphism In the case of addition, we can decrypt as

$$(c_1+c_2)(u_x(t),u_y(t)) = m_1(t)+m_2(t)+\ell\cdot(e_1+e_2)(u_x(t),u_y(t)) \tag{79}$$

and extract the plaintext

$$(c_1+c_2)(u_x(t),u_y(t))\ (mod\ \ell) = m_1(t)+m_2(t) \tag{80}$$

under the following conditions.

$$MC(m_1(t)+m_2(t)) < \ell \tag{81}$$

$$\ell\cdot MC((e_1+e_2)(u_x(t),u_y(t))) < q \tag{82}$$

The condition (81) is to prevent the coefficients of the plaintext $m_1(t)+m_2(t)$ from overflowing beyond the range 0 to $q-1$. The condition (82) is to prevent the coefficients of the noise term $\ell\cdot(e_1+e_2)(u_x(t),u_y(t))$ from overflowing beyond the range of 0 to $q-1$.

Let N_a be the number of times to add ciphertext. Then we obtain the condition of ℓ and q as follows:

$$\ell > N_a\cdot\lambda, \tag{83}$$

$$q > N_a\cdot(\ell-1+\ell\sum_{k=0}^{dX+dr}(k+1)n^k\dot{(}\ell-1)^{k+1}), \tag{84}$$

where λ is a parameter giving the maximum size of the coefficients in m, $u_x(t)$ and $u_y(t)$ in the case of using the homomorphic operations. So, we can perform additional homomorphic operation between $n\log_2\ell$ bit integers.

Multiplicative homomorphism We can multiply

$$\begin{aligned}(c_1c_2)(u_x(t),u_y(t)) = m_1(t)m_2(t)+\ell\cdot(&m_1(t)e_2(u_x(t),u_y(t))\\ &+m_2(t)e_1(u_x(t),u_y(t))+\ell\cdot e_1(u_x(t),u_y(t))e_2(u_x(t),u_y(t)))\end{aligned}$$

and extract the plaintext

$$(c_1c_2)(u_x(t),u_y(t))\ (mod\ \ell) = m_1(t)m_2(t) \tag{85}$$

under the following conditions.

$$MC(m_1(t)m_2(t)) < \ell \tag{86}$$

$$\deg m_1(t)m_2(t) < n \tag{87}$$

$$\ell^2\cdot MC((e_1e_2)(u_x(t),u_y(t))) < q \tag{88}$$

The condition (86) is to prevent the coefficients of the plaintext $m_1(t)m_2(t)$ from overflowing beyond the range 0 to $\ell - 1$. Also, $m_1(t)m_2(t)$ requires keeping the degree under n since $m_1(t)m_2(t)$ must be included in R_ℓ, as required by the condition (87).

The condition (88) is to prevent the coefficients of the noise term from overflowing beyond the range 0 to $q - 1$ after the multiplication.

Let N_m be the number of times to multiply ciphertext. Then, we obtain the conditions for ℓ and n as follows.

$$\lambda^{N_m+1} < \ell \tag{89}$$

$$N_m \deg m(t) < n \tag{90}$$

We can therefore perform N_m multiplicative homomorphic operations between $n \log_2 \lambda / N_m$ bit integers. The condition for q can be calculated recursively by applying the discussion for the condition (15) in the Theorem 1.

12 Conclusion

In this study, we constructed a post-quantum encryption scheme whose security is based on an IE-LWE problem and related to the small-solution problem in non-linear spaces. This paper gave the algorithms for key generation, encryption/decryption, and the security proof in the sense of IND-CPA. Then, we discussed two attacks that can be applied to the IE-LWE problem and concluded the key recovery attack is dominant of them in the case of $\deg X(x, y) = 1$. We precisely investigated the lattice that is associated with the key recovery attack and estimated an appropriate parameters of our scheme according to the "2016 estimate" which is a reliable method to estimate the computational complexity of the lattice reduction. We are going to estimate appropriate parameters for $\deg X(x, y) > 1$.

Acknowledgments

The authors thank Keita Xagawa for suggesting us the attack [6] and [22] which may work against our scheme when we choose the parameter n to be composite. The authors also thank Ward Beullens, Wouter Castryck and Frederik Vercauteren for pointing out the evaluating at one attack can be applied to our scheme and proposing an indicator called "distinguishing advantage" to see the bias in the distribution of the ciphertext [14] and [51]. Phong Nguyen suggests the security was overestimated due to our non-optimized attack [37]. This paper has revised based on their suggestions.

References

1. M. Ajtai, C. Dwork, *A Public-Key CryptosystemwithWorst-Case/Average-Case Equivalence*, In: Proc. of STOC'97, ACM, pp. 284–293, ACM New York, NY, USA, (1997).

2. K. Akiyama, Y. Goto, *A Public-key Cryptosystem using Algebraic Surfaces*, In: Proc. of PQCrypto'06, pp. 119–138, (2006), available at `http://postquantum.cr.yp.to/`.
3. K. Akiyama, Y. Goto, H. Miyake, *An Algebraic Surfaces Cryptosytem*, In: Proc. of PKC'09, **5443**, Lecture Notes in Computer Science, pp. 425–442, Springer Berlin Heidelberg, (2009).
4. K. Akiyama, Y. Goto, S. Okumura, T. Takagi, K. Nuida, G. Hanaoka, *A Public-key Encryption Scheme Based on Non-linear Indeterminate Equations*, In: Proc. of SAC 2017, **10719**, Lecture Notes in Computer Science, pp. 215-234, Springer Berlin Heidelberg, (2018).
5. K. Akiyama, Y. Goto, S. Okumura, T. Takagi, K. Nuida, G. Hanaoka, H. Shimizu, Y. Ikematsu, *Indeterminate Equation Public-key Cryptosystem (Giophantus^{TM})*, In the round-1-submissions of NIST PQC standardization, (2017), available at `https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions`,
6. M. R. Albrecht, S. Bai; Leo Ducas, *A subfield lattice attack on overstretched NTRU assumptions*, In: Proc. of CRYPTO'16, **9814–9816**, Lecture Notes in Computer Science, pp. 153–178, Springer Berlin Heidelberg, (2016).
7. M. Albrecht, F.Göpfert, F.Virdia, T.Wunderer, *Revisiting the Expected Cost of Solving uSVP and Application to LWE*, In: Proc. of ASIACRYPT'17, **10624**, Lecture Notes in Computer Science, pp. 297-322, Springer Berlin Heidelberg, (2017).
8. M. Albrecht, B. Curtis, A. Deo. A. Davidson, R. Player, E. Postlethwaite, F. Virdia, Thomas Wunderer, *Estimate all the {LWE, NTRU} schemes!*, `https://estimate-all-the-lwe-ntru-schemes.github.io/docs/`, (2018).
9. E. Alkim, L. Ducas, T. Pöppelmann, P.Schwabe, *Post-quantum Key Exchange - A New Hope*, 25th USENIX Security Symposium (USENIX Security 16), pp.327–343,(2016).
10. S. Bai, S. D. Galbraith, *Lattice Decoding Attacks on Binary LWE*, ACISP'14, **8544**, Lecture Notes in Computer Science, pp. 322–337, Springer International Publishing, (2014).
11. A. Becker, L. Ducas, N. Gama, T. Laarhoven,"New directions in nearest neighbor searching with applications to lattice sieving", https://eprint.iacr.org/2015/1128.pdf
12. D. Boneh, E.-J. Goh, K. Nissim, *Evaluating 2-DNF Formulas on Ciphertexts*, In: Proc. of TCC'05, **3378**, Lecture Notes in Computer Science, pp.325–341, Springer Berlin Heidelberg, (2005).
13. Z. Brakerski, A. Langlois, C. Peikert, O. Regev, D. Stehlé, *Classical hardness of learning with errors*, In: Proc. of STOC'13, ACM, pp. 575–584, ACM New York, NY, USA, (2013).
14. W. Beullens, W. Castryck, F. Vercauteren, *IND-CPA attack on Giophantus*, In the official-comments to Giophantus for NIST Round-1-submissions , (2018)
15. Y. Chen, P. Nguyen, *BKZ 2.0: Better lattice security estimates*, In: Proc. of ASIACRYPT'11, **7073**, Lecture Notes in Computer Science, pp. 1–20, Springer Berlin Heidelberg, (2011).
16. Y. Chen, *Reduction de reseau et securite concrete du chirement completement homomorphe* PhD thesis, Paris 7, (2013).
17. J. Denef, *The Diophantine Problem for Polynomial Rings of Positive Characteristic*, In: Proc. of Logic Colloquium '78, Studies in Logic and the Foundations of Mathematics, **97**, pp. 131–145, North Holland, Amsterdam-New York, (1979).
18. T. ElGamal, *A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*, In: Proc. of CRYPTO'84, **196**, Lecture Notes in Computer Science, pp. 10–18, Springer Berlin Heidelberg, (1984).

19. J. Faugère, P. Spaenlehauer, *Algebraic Cryptanalysis of the PKC'09 Algebraic Surface Cryptosystem*, In: Proc. of PKC'10, **6056**, Lecture Notes in Computer Science, pp. 35–52, Springer Berlin Heidelberg, (2010).
20. E. Fujisaki, T. Okamoto, "How to Enhance the Security of Public-Key Encryption at Minimum Cost,"In: Proc. of PKC'99, **1560**, Lecture Notes in Computer Science, pp.53–68, Springer Berlin Heidelberg, (1999).
21. N. Gama, P. Q. Nguyen, *Predicting lattice reduction*, In: Proc. of EUROCRYPT'08, **4963**, Lecture Notes in Computer Science, pp. 31–51, Springer Berlin Heidelberg, (2008).
22. C. Gentry, *Key Recovery and Message Attacks on NTRU-Composite*, In: Proc. of EUROCRYPT'01, **2045**, Lecture Notes in Computer Science, pp. 182–194 Springer Berlin Heidelberg, (2001).
23. C. Gentry, *Fully Homomorphic Encryption Using Ideal Lattices*, In: Proc. of STOC'09, ACM, pp.169–178, ACM New York, NY, USA, (2009).
24. O. Goldreich, S. Goldwasser, S. Halevi, *Public-Key Cryptosystems from Lattice Reduction Problems*, In: Proc. of CRYPTO'97, **1294**, Lecture Notes in Computer Science, pp.112–131, Springer Berlin Heidelberg, (1997).
25. J. Hoffstein, J. Pipher, J. H. Silverman, *NTRU: A Ring-Based Public Key Cryptosystem*, In: Proc. of ANTS'98, **1423**, Lecture Notes in Computer Science, Springer-Verlag, pp.267–288, Springer Berlin Heidelberg, (1998).
26. H. Imai, T. Matsumoto, *Public quadratic polynomial-tuples for efficient signature-verification and message-encryption*, In: Proc. of EUROCRYPT'88, **330**, Lecture Notes in Computer Science, pp. 419-453, Springer Berlin Heidelberg, (1989).
27. R. Kannan, *Minkowski's convex body theorem and integer programming*, Math. of Oper. Res., **12**(3), pp. 415–440, INFORMS, Linthicum, Maryland, USA, (1987).
28. P. Kirchner, P.-A. Fouque, *Comparison between Subfield and Straightforward Attacks on NTRU*, IACR Cryptology ePrint Archive: Report 2016/717, available at `http://eprint.iacr.org/2016/717`.
29. K. Kobara, H. Imai, *Semantically secure McEliece public-key cryptosystems - conversion for McEliece PKC*, In: Proc. of PKC'01, **1992**, Lecture Notes in Computer Science, pp.19–35, Springer Berlin Heidelberg, (2001).
30. A. K. Lenstra, H. W. Lenstra, L. Jr.Lovasz, *Factoring polynomials with rational coefficients*, Math. Ann. **261**(4), pp. 515–534, Springer-Verlag, (1982).
31. R. Lindner, C. Peikert, *Better key sizes (and attacks) for LWE-based encryption*, In: Proc. of CT-RSA'11, **6558**, Lecture Notes in Computer Science, pp. 319–339, Springer Berlin Heidelberg, (2011).
32. M. Liu, P. Q. Nguyen *Solving BDD by Enumeration: An Update*, In: Proc. of CT-RSA'13, **7779**, Lecture Notes in Computer Science, pp. 293–309, Springer Berlin Heidelberg, (2013).
33. R. J. McEliece, *A Public-Key Cryptosystem Based On Algebraic Coding Theory*, The Deep Space Network Progress Report, DSN PR, 42-44, pp.114–116, (1978).
34. D. Micciancio, C. Peikert, *Hardness of SIS and LWE with Small Parameters*, In: Proc. of CRYPTO'13, **8042**, Lecture Notes in Computer Science, pp. 21–39, Springer Berlin Heidelberg, (2013).
35. P. Nguyen, *Cryptanalysis of the Goldreich-Goldwasser-Halevi Cryptosystem from Crypto '97*, Proc. of CRYPTO'99, **1666**, Lecture Notes in Computer Science, pp. 288–304, Springer Berlin Heidelberg, (1999).
36. P. Nguyen, J. Stern, *Cryptanalysis of the Ajtai-Dwork Cryptosystem*, In: Proc. of CRYPTO'98, **1462**, Lecture Notes in Computer Science, pp.223–242, Springer Berlin Heidelberg, (1998).

37. P. Nguyen, *Giophantus and *LWR-based submissions*, In the official-comments to Giophantus for NIST Round-1-submissions, (2018).
38. R. Niebuhr, M. Meziani, S. Bulygin, J. Buchmann, *Selecting parameters for secure McEliece-based cryptosystems*, Int. J. Inf. Secur., **11**(3), pp. 137-147, Springer-Verlag, (2012).
39. H. Niederreiter, *Knapsack-type cryptosystems and algebraic coding theory*, Probl. Control Inform. Theory/Problemy Upravlen. Teor. Inform., **15**(2), pp.19–34, (1986).
40. NIST, *NIST Post-Quantum Cryptography Standardization*, `https://csrc.nist.gov/Projects/Post-Quantum-Cryptography`, (2016).
41. K. Nuida, K. Kurosawa, *(Batch) Fully Homomorphic Encryption over Integers for Non-Binary Message Spaces*, In: Proc. of EUROCRYPT'15, **9056**, Lecture Notes in Computer Science, pp.537–555, Springer Berlin Heidelberg, (2015).
42. P. Paillier, *Public-Key Cryptosystems Based on Composite Degree Residuosity Classes*, In: Proc. of EUROCRYPT'99, **1592**, Lecture Notes in Computer Science, pp. 223–238, Springer Berlin Heidelberg, (1999).
43. J. Patarin, *Hidden Field Equations (HFE) and Isomorphisms of Polynomials(IP): Two New Families of Asymmetric Algorithms*, In: Proc. of EUROCRYPTO'96, **1070**, Lecture Notes in Computer Science, pp. 33-48, Springer Berlin Heidelberg, (1996).
44. J. Porras, J. Baena, J. Ding, *ZHFE, a new multivariate public key encryption scheme*, In: Proc. of PQCrypto'14, **8772**, Lecture Notes in Computer Science, pp. 229–245, Springer International Publishing, (2014).
45. O. Regev, *On Lattices, Learning with Errors, Random Linear Codes, and Cryptography*, J. of the ACM, **56**(6), pp. 1–40, ACM New York, NY, USA, (2009).
46. C. R. Schnorr, M. Euchner, *Lattice basis reduction: improved algorithms and solving subset sum problems*, in Math. and Programming, **66**(1), pp. 181-189, Springer-Verlag, (1994).
47. P. W. Shor, *Algorithms for Quantum Computation: Discrete Log and Factoring*, In: Proc. of SFCS'94, pp. 124–134, IEEE Computer Society Washington, (1994).
48. D. Stehle, R. Steinfeld, *Making NTRU as secure as worst-case problems over ideal lattices*, In: Proc. of EUROCRYPT'11, **6632**, Lecture Notes in Computer Science, pp. 27–47, Springer Berlin Heidelberg, (2011).
49. C. Tao, A. Diene, S. Tang, J. Ding, *Simple matrix scheme for encryption*, In: Proc. of PQCrypto'13, **7932**, Lecture Notes in Computer Science, pp.231–242, Springer Berlin Heidelberg, (2013).
50. W. Castryck, I. Iliashenkom, F.Vercauteren, *Provably Weak Instances of Ring-LWE Revisited*, In: Proc. of EUROCRYPT'16, **9665**, Lecture Notes in Computer Science, pp.147–167, Springer Berlin Heidelberg, (2016).
51. F.Vercauteren, *OFFICIAL COMMENT: Giophantus*, In the official-comments to Giophantus for NIST Round-1-submissions, (2018)
52. K. Xagawa, *Practical Cryptanalysis of a Public-Key Encryption Scheme Based on Non-linear Indeterminate Equations at SAC 2017*, In: Proc. of PQCrypto'18, **10786**, Lecture Notes in Computer Science, pp.142–161, Springer Berlin Heidelberg, (2018).
53. T. Yasuda, K. Sakurai, *A multivariate encryption scheme with Rainbow*, In: Proc. of ICICS'15, **9543**, Lecture Notes in Computer Science, pp.222–236, Springer International Publishing, (2016).