QTRU: A Lattice Attack Resistant Version of NTRU PKCS Based on Quaternion Algebra

Ehsan Malekian[1], Ali Zakerolhosseini[1]*, Atefeh Mashatan[2]
[1] Faculty of Electrical & Computer Engineering
Shahid Beheshti University, Tehran, Iran
[2] Security and Cryptography Laboratory (LASEC) - EPFL
CH-1015, Lausanne, Switzerland

April 13, 2009

Abstract

We propose QTRU, a probabilistic and multi-dimensional public-key cryptosystem based on the NTRU public-key cryptosystem using quaternion algebra. QTRU encrypts four data vectors in each encryption session and the only other major difference between NTRU and QTRU is that the underlying algebraic structure has been changed to a non-commutative algebraic structure. As a result, QTRU inherits the strength of NTRU and its positive points. In addition, the non-commutativity of the underlying structure of QTRU makes it much more resistant to some lattice-based attacks.

After a brief description of NRTU, we begin by describing the algebraic structure used in QTRU. Further, we present the details of the key generation, encryption and decryption algorithms of QTRU and discuss the issues regarding key security, message security, and probability of successful decryption. Last but not least, QTRU's resistance against lattice-based attacks is investigated.

Keywords: QTRU, NTRU, quaternion algebra, public-key cryptography, encryption

1 Introduction

NTRU is a probabilistic public-key cryptosystem that was first proposed by Jeffrey Hoffstein, Jill Pipher and Joseph H. Silverman in the rump session of Crypto' 96 and the first official paper was published in 1998 [1]. Compared to more well-known systems such as RSA or ECC, the greatest advantage of NTRU is that it is based on a class of basic arithmetic operations whose inherent complexity is rather low, amounting to $\mathcal{O}(N^2)$ in worst-case. Hence, NTRU is considered to be computationally inexpensive. Computational efficiency along with low cost of implementation have

*Corresponding author.

turned NTRU into a very suitable choice for a large number of applications such as embedded systems, mobile phones, portable devices and resource constrained devices [2].

As a rough comparison, NTRU is hundreds of times faster than RSA and has a much faster key generation algorithm. However, there is an obvious drawback in using NTRU in that sometimes the decryption process fails to give the plaintext back. There have been some suggestions in the literature to overcome this problem, check error & re-encrypt, for instance. **000 this needs a reference**

NTRU is classified as a lattice-based cryptosystem since it has taken its inherent security from intractability of hard-problems in certain types of lattices, contrary to RSA and ECC. On the other hand, NTRU is also classified as a probabilistic cryptosystem as each encryption involves a random vector and, hence, messages do not have unique encryptions.

During the past ten years, NTRU has been meticulously analyzed by researchers and its main core is still assumed to be safe. Most sophisticated attacks against NTRU are based on lattice reduction techniques. Two famous lattice problems, Shortest Vector Problem (SVP) and Closest Vector Problem (CVP), have shown to be among NP-hard problems. **000 this claim needs reference** However, the lattice problem arising in NTRU is classified as a Convolution Modular Lattice (CML) and it is not determined yet whether or not the cyclic structure of CML is going to help reducing the complexity of CVP or SVP. This issue has been considered in new versions of NTRU [3, 4]. Yet, some open problems with regard to CML remain. **000 give reference to these problems, or explain them, or better just delete this last sentence**

In this paper, we present QTRU which is a cryptosystem based on NTRU while the underlying algebra is quaternion algebra. As a result of non-commutativity of the underlying algebraic structure, and bi-linearity of multiplication, many lattice reduction algorithms will not work. Consequently, we can reduce the dimension of the vector space considerably and, yet, obtain the same level of security.

In completely even circumstances, i.e., choosing the same parameters for both NTRU and QTRU, QTRU works four times slower than NTRU and the data are encrypted simultaneously as four vectors. Other than changing the underlying algebra, no other change as been made. In particular, QTRU keeps the probabilistic properties of NTRU. Hence, QTRU inherits the main advantages of NTRU.

Since four vectors of data are encrypted simultaneously in each system call, QTRU can be considered as a multidimensional cryptosystem. As a result of high complexity of the lattice which arises from QTRU, the dimension of the cryptosystem lattice can be reduced and will arguably compensate the imposed speed reduction caused by encrypting the four vectors.

This paper is organized as follows: Section 2 summarizes the NTRU cryptography system. Then, Section 3 includes a brief introduction to quaternion algebras. We dedicate Section 4 to introducing the algebraic structure used in QTRU. Then, Sections 5 and 6 are devoted to the description of QTRU and general analysis of the scheme. Last but not least, 7 discusses the security of QTRU against lattice based attacks.

2 NTRU Cryptosystem

The basic operations in NTRU take place in the ring $\mathbb{Z}[x]/(x^N - 1)$, which is known as the convolution ring, where N is a prime [5]. In the convolution ring, addition and multiplication have complexity $\mathcal{O}(N)$ and $\mathcal{O}(N^2)$, respectively. Hence, the selection of this ring as the algebraic structure of NTRU provides us the associated speed and efficiency.

Following the notation of [5] and [6], we define the following three rings: $\mathcal{R} \doteq \mathbb{Z}[x]/(x^N - 1)$, $\mathcal{R}_p \doteq (\mathbb{Z}/\mathbb{Z}_p)[x]/(x^N - 1)$, and $\mathcal{R}_q \doteq (\mathbb{Z}/\mathbb{Z}_q)[x]/(x^N - 1)$.

An element f from any of the three rings $\mathcal{R}$, $\mathcal{R}_p$, and $\mathcal{R}_q$, can be written as a polynomial or a vector of coefficients: $f = \sum_{i=0}^{N-1} f_i.x_i \doteq [f_0, f_1, ..., f_{N-1}]$

Addition is the ordinary addition for polynomials, i.e., element-wise vector addition, but multiplication, denoted by $\star$ is explicitly defined as:

$$f \star g = h$$

where:

$$h_k = \sum_{i=0}^{k} f_i.g_{k-i} + \sum_{i=k+1}^{N-1} f_i.g_{N+k-i} = \sum_{i+j \overset{N}{\equiv} k} f_i.g_i$$

Clearly, addition and multiplication in $\mathcal{R}_p$ or $\mathcal{R}_q$ are equivalent to performing the same operations in $\mathcal{R}$ and ultimately reducing the resulting coefficients mod p or mod q.

Let d_f, d_g, d_ϕ, and d_m be constant integers less than N which are public parameters of the cryptosystem and determine the distribution of the coefficients of the polynomials. Based on these constants, we shall define the subsets $\mathcal{L}_f$, $\mathcal{L}_m$, $\mathcal{L}_\phi$, $\mathcal{L}_g \subset \mathcal{R}$ based on the criteria presented in Table (1).

Notation	***Definition***	***Typical Value for N=167, p=3, q=128***
$\mathcal{L}_f$	$\{f \in \mathcal{R} \vert f$ has d_f coefficients equal to +1, $(d_f - 1)$ equal to -1, the rest 0$\}$	d_f =61
$\mathcal{L}_g$	$\{g \in \mathcal{R} \vert g$ has d_g coefficients equal to +1, d_g equal to -1, the rest 0$\}$	d_g =20
$\mathcal{L}_\phi$	$\{\phi \in \mathcal{R} \vert \phi$ has d_ϕ coefficients equal to +1, d_ϕ equal to -1, the rest 0$\}$	d_ϕ =18
$\mathcal{L}_m$	$\{$ $m \in \mathcal{R} \vert$ coefficients of m are chosen modulo p, between $-p/2$ and $p/2\}$	-

Table 1: Definition of public parameters of NTRU

Having set the above parameters, the NTRU cryptosystem can now be described as follows:

Public Parameters The following parameters in NTRU are assumed fixed and public and must be agreed upon by both sender and receiver:

- N is a prime number which determines the structure of the ring $\mathbb{Z}[x]/(x^N - 1)$ and its value has considerable effects on the system's security and speed. (Generic values for N include $N = 107$ for moderate security, $N = 167$ for high security, and $N = 503$ for very high security)

- p and q are two numbers which are relatively prime and q is much greater than p. (Typical values: $p = 3$, and $q = 64, 128, 256$)

- d_f, d_g, d_ϕ, and d_m are constant parameters as defined in Table (1). These constants have great impact on the rate of decryption failure as well as the system security.

Key Generation To create an NTRU key, first two small polynomials $g \in \mathcal{L}_g$ and $f \in \mathcal{L}_f$ are randomly generated. The polynomial f must be invertible in the rings $\mathcal{R}_p$ and $\mathcal{R}_q$. When f is randomly selected from the subset $\mathcal{L}_f$, the probability for this polynomial to be invertible in $\mathcal{R}_p$ and $\mathcal{R}_q$ will be very high, however, in rare event that f is not invertible, a new polynomial f can be easily generated.

If q is a power of a prime s, that is $q = s^k$, then one can count the number of irreducible polynomials in $\mathcal{R}_q$. Note that f is chosen in a way that it is never divided by $(x - 1)$. Hence, the probability that f is invertible over $\mathcal{R}_q$ is [5] about $(1 - p^{-n})^{(N-1)}/n$, where n is the smallest integer which satisfies $p^n = 1 \bmod N$.

The inverse of f over $\mathcal{R}_p$ and $\mathcal{R}_q$ is computed using the extended Euclidian algorithm. We call those two inverses f_p^{-1} and f_q^{-1}, respectively. Hence, we have $f_p^{-1} \star f \equiv 1 \pmod{p}$ and $f_q^{-1} \star f \equiv 1 \pmod{q}$).

While f, g, f_p^{-1}, and f_q^{-1} are kept private, the public-key h is computed in the following manner:

$$h = f_q^{-1} \star g \pmod{q}.$$

Encryption For encryption, the system initially selects a random polynomial $\phi \in \mathcal{L}_\phi$, called the blinding polynomial, and converts the input message to a polynomial $m \in \mathcal{L}_m$. The ciphertext is computed as follows:

$$e = p.h \star \phi + m \ (mod\ q).$$

Note that p is a constant parameter and we can pre-compute the polynomial $p.h$. Hence, regardless of the time required for generating the blinding polynomial and transforming the incoming message into the polynomial m, encryption process demands N^2 multiplication and N addition $\bmod q$. With the selection of $q = 2^l$, the cost of coefficients' reduction $\bmod q$ equals zero.

Decryption In order to decrypt, the received polynomial e is multiplied (convoluted) by the private key f:

$$f \star e \pmod{q} = f \star (p.h \star \phi + m) \pmod{q}$$

$$= p.f \star h \star \phi + f \star m \pmod{q}$$

$$= p.f \star f_q^{-1} \star g \star \phi + f \star m \pmod q$$

$$= p.g \star \phi + f \star m \pmod q.$$

Through suitable selection of system parameters, the coefficients of the polynomial $p.g \star \phi + f \star m$ will most probably lie in the interval $(-q/2, +q/2]$ and there will be no need for reduction mod$q$. With this assumption, when we reduce the result of $p.g \star \phi + f \star m$ by *mod p*, the term $p.g \star \phi$ will vanish and $f \star m$ (mod$p$) will remain. In order to extract the message m, it is enough to multiply $f \star m$ (mod$p$) by f_p^{-1} and then adjust the resulting coefficients within the interval $[-p/2, +p/2)$. Given this description, the decryption process includes two convolution multiplications and hence the decryption speed is less than half of the encryption speed.

Successful Decryption Consider $f \in \mathcal{R}$ as a polynomial with coefficient vector $[f_0, f_1, ..., f_{N-1}]$. Then, *width* of f, denoted by $|f|_\infty$, is defined as follows:

$$|f|_\infty = \max_{0 \le i \le N-1}(f_i) - \min_{0 \le i \le N-1}(f_i) \ .$$

Now, the probability of successful decryption is approximately determined with a few simplifying assumptions:

(1) g_i's and f_i's are independent random variables,

(2) coefficients of $f \star m = \sum_{i+j \overset{N}{\equiv} k} f_i.m_i$ and $g \star \phi = \sum_{i+j \overset{N}{\equiv} k} g_i.\phi_i$ have normal distribution around zero, and

(3) $\Pr(m_i = 1) = \Pr(m_i = -1) = \Pr(m_i = 0) = \frac{1}{3}$.

Successful decryption depends on whether or not $|p.g \star \phi + f \star m|_\infty < q$. Through a few simple probabilistic calculations [6], the approximate bound for successful decryption probability can be calculated as follows.

$$\Pr(successful\ decryption) = \left(2\Phi(\frac{q-1}{2\sigma}) - 1\right)^N ,$$

where Φ denotes the distribution of the standard normal variable and $\sigma \approx \sqrt{\frac{36 d_f . d_g}{N} + \frac{8 d_f}{6}}$.

The numerical results show that for typical values, e.g., $N = 167$, $p = 3$, $q = 128$, $d_f = 61$, $d_g = 20$, and $d_\phi = 18$, the probability of failure in decryption is about 10^{-5}.

Lattice Attacks against NTRU The hard underlying problem of NTRU is to find short vectors in Convolution Modular Lattices (CML) [7]. There have been many papers on lattice attacks against NTRU [3, 4, 5, 7, 8, 9, 10].

Consider the public-key h as a vector $h = [h_0, h_1, ..., h_{N-1}]$. Then, the standard NTRU lattice with dimension $2N$ is generated by rows of the following matrix:

$$\mathcal{L}_{NTRU} = \left[\begin{array}{c|c} \lambda.I & h \\ \hline 0 & q.I \end{array}\right]$$

$$= \left[\begin{array}{ccccc|ccccc} \lambda & 0 & 0 & \cdots & 0 & h_0 & h_1 & h_2 & \cdots & h_{N-1} \\ 0 & \lambda & & & 0 & h_{N-1} & h_0 & h_3 & \cdots & h_{N-2} \\ 0 & & \ddots & & & h_{N-2} & h_{N-1} & & & h_{N-3} \\ \vdots & & & & \vdots & \vdots & \vdots & & \ddots & \vdots \\ 0 & & \cdots & & \lambda & h_1 & h_2 & & \cdots & h_0 \\ \hline 0 & 0 & & \cdots & 0 & q & 0 & 0 & \cdots & 0 \\ 0 & 0 & & & 0 & 0 & q & & & \\ & & \ddots & & & 0 & & & & \\ \vdots & & & & \vdots & \vdots & & & \ddots & \vdots \\ 0 & & & \cdots & 0 & 0 & & & \cdots & q \end{array}\right].$$

We can assume that the parameter λ, known as the balancing constant, is equal to 1. Typically, λ is selected in a way that makes the search for short vectors in CML more efficient. According to [9], the best choice for λ is a value around $\|f\| \,/\, \|g\|$. With regards to the public information about f and g, we know that the following vector is in $\mathcal{L}_{NTRU}$ with a relatively small norm:

$$v = (\lambda f_0, \lambda f_1, \cdots, \lambda f_{N-1}, g_0, g_1, \cdots g_{N-1}).$$

An attacker tries to search for short vectors having norm around v, using formation of such a lattice and lattice reduction algorithms. Consider $f^{(k)}$ as the symbol for cyclic shift in the vector f with k shifts. The difference between CML and other lattice types is that if the vector $v = (\lambda f, g)$ exists in the lattice, then the entire N shifted vectors will also have the same norm and will be in the lattice $\mathcal{L}_{NTRU}$.

On the other hand, f and g in the vector coordinates possess $N - 2.d_f$ and $N - 2.d_g$ zero elements, respectively. A method has been presented in [7] based on which, this property (runs of zeros in f and g) can be utilized for lattice dimension reduction. However, apparently even by the use of all properties of CML and reliance upon several tricks introduced in [10], as well as using the best variation of the lattice reduction algorithms, only NTRU-107 (N=107) is breakable. The estimated bit-security for NTRU-167 version is nearly 57 bits, 83 bits for NTRU-251 and 180 bits for NTRU-503 which appears to suffice for various applications in real world scenarios [3, 4, 9, 10].

3 A Brief Introduction to Quaternion Algebra

In this section, quaternion algebra is briefly introduced. Readers can refer to Conway's book [11] or [12, 13] for a more elaborate description. Real quaternion , denoted by $\mathbb{H}$, can be regarded as a vector space of dimension 4 over $\mathbb{R}$. Quaternion algebra, discovered by Sir William Rowan Hamilton in 1843, is the second normed division algebra in the sense of Cayley-Dickson construction method. By *algebra* it means a vector space V over $\mathbb{R}$ (or generally over any field $\mathbb{F}$) that is equipped with a bilinear map. An algebra $\mathbb{A}$ is called *division algebra* provided that for every a,$b \in \mathbb{A}$, $a.b = 0$ implies $a = 0$ or $b = 0$. In other words, division algebra does not have any zero divisors. *Normed division algebra* is a division algebra equipped with a multiplicative norm function, denoted by $\|.\|$. A normed division algebra is not necessarily commutative or associative. Typically, the elements of $\mathbb{H}$ are denoted by the expression $\alpha + \beta.i + \gamma.j + \delta.k$, where $\alpha, \beta, \gamma, \delta \in \mathbb{R}$. That is,

$$\mathbb{H} = \{\alpha + \beta.i + \gamma.j + \delta.k | \alpha, \beta, \gamma, \delta \in \mathbb{R}\}.$$

A quaternion can be shown by ordinary vector notations $q =< \alpha, \beta, \gamma, \delta >$ over $\mathbb{R}^4$ or by $q =< \alpha, \beta >$ over $\mathbb{C}^2$ when there us no ambiguity. As a vector space, addition and scalar multiplication are defined by ordinary element-wise vector addition and scalar multiplication, but multiplication of two quaternions shall be done according to the following rules:

$$i^2 = j^2 = k^2 = -1 \text{ and } ij = -ji = k.$$

The set of real quaternions together with ordinary addition and multiplication defined as above, forms a *skew field* [14]. For each quaternion $q =< \alpha, \beta, \gamma, \delta >$, the conjugate, denoted by $\bar{q}$, is given by $\bar{q} = \alpha - \beta.i - \gamma.j - \delta.k$, and the norm is defined by $N(q) = q \times \bar{q} = \bar{q} \times q = \alpha^2 + \beta^2 + \gamma^2 + \delta^2$. The inverse of the quaternion q is defined by $q^{-1} = \frac{\bar{q}}{N(q)}$, provided that it has a nonzero norm ($N(q) \neq 0$). The set of all real quaternions with norm 1, forms a non-commutative multiplicative group known as $SU(2)$ that is isomorphic to multiplicative group of all 2×2 matrices of determinant 1 over $\mathbb{C}$.

Quaternion algebra can be generalized by replacing the field of real numbers $\mathbb{R}$ by any arbitrary field $\mathbb{F}$ (or ring $\mathcal{R}$). Moreover, instead of defining $i^2 = j^2 = k^2 = -1$ and $ij = -ji = k$, one can define i, j and k as $i^2 = a$, $j^2 = b$, $k^2 = -ab$ and $ij = -ji = k$. This will achieve a general non-commutative algebraic system.

Assume $\mathbb{F}$ is an arbitrary field and the characteristic of $\mathbb{F}$ is not 2. Then, the quaternion algebra $\mathbb{A}$ can be defined over $\mathbb{F}$ as:

$$\mathbb{A} \doteq \left(\frac{a, b}{F}\right) \doteq$$

$$\left\{\alpha + \beta.i + \gamma.j + \delta.k \,\middle|\, \alpha, \beta, \gamma, \delta \in \mathbb{F},\ i^2 = a, j^2 = b, ij = -ji = k\right\}.$$

Clearly, if we let a and b to be -1 and $\mathbb{F}$ to be the field of real numbers $\mathbb{R}$, we obtain the Hamiltonian quaternion, i.e., $\mathbb{H} = \left(\frac{-1,-1}{\mathbb{R}}\right)$. Based on the choice of a and b and the nature of the field $\mathbb{F}$, $\mathbb{A} = \left(\frac{a,b}{\mathbb{F}}\right)$, we get two different isomorphism types:

1. $\mathbb{A} = \left(\frac{a,b}{\mathbb{F}}\right)$ is an Euclidean division ring (skew field) if and only if for $q \in \left(\frac{a,b}{\mathbb{F}}\right)$, $N(q) = 0$ results in $q = 0$. This property demands that according to the definition $q^{-1} = \frac{\bar{q}}{N(q)}$, existence of the inverse can be guaranteed for all non-zero elements and hence quaternion algebra possess all conditions to be a skew field and normed division algebra.

2. $\mathbb{A} = \left(\frac{a,b}{\mathbb{F}}\right)$ is isomorphic to $M_2(\mathbb{F})$, the ring of all 2×2 matrices with entries from $\mathbb{F}$. Such an algebra is called a *split* algebra. In an split algebra, there are some nonzero elements $q \in \mathbb{A}$ which have no multiplicative inverses. Assuming $\mathbb{F} = GF(p)$ or $\mathbb{F} = GF(p^n)$, algebra $\mathbb{A} = \left(\frac{a,b}{\mathbb{F}}\right)$ is absolutely a split algebra [12, 13].

4 Algebraic Structure of the Proposed Scheme

Consider the two rings $\mathbb{Z}_p[x]/(x^N - 1)$ and $\mathbb{Z}_q[x]/(x^N - 1)$ that are used in NTRU. We define two quaternionic algebras $\mathbb{A}_0$ and $\mathbb{A}_1$ as follows:

$$\mathbb{A}_0 = \left(\frac{-1, -1}{\mathbb{Z}_p[x]/(x^N - 1)}\right),$$

$$\mathbb{A}_1 = \left(\frac{-1, -1}{\mathbb{Z}_q[x]/(x^N - 1)}\right).$$

For simplicity, p, q and N are assumed to be prime numbers. Since $\mathbb{Z}_p[x]/(x^N - 1)$ and $\mathbb{Z}_q[x]/(x^N - 1)$ are finite rings with characteristics p and q, respectively, one can easily conclude that $\mathbb{A}_0$ and $\mathbb{A}_1$ algebras are split. In other words, $\mathbb{A}_0$ and $\mathbb{A}_1$ algebras possess all characteristics of quaternion algebras, except that there are some nonzero elements whose norm is zero and naturally such elements do not have a multiplicative inverse. Let's elaborate more on algebras $\mathbb{A}_0$ and $\mathbb{A}_1$:

$$\mathbb{A}_0 = \left(\frac{-1, -1}{\mathbb{Z}_p[x]/(x^N - 1)}\right) =$$

$$\left\{ f_0(x) + f_1(x).i + f_2(x).j + f_3(x).k \,\middle|\, f_0, f_1, f_2, f_3 \in \mathbb{Z}_p[x]/(x^N - 1),\ i^2 = -1, j^2 = -1, ij = -ji = k \right\}.$$

$$\mathbb{A}_1 = \left(\frac{-1, -1}{\mathbb{Z}_q[x]/(x^N - 1)}\right) =$$

$$\left\{ g_0(x) + g_1(x).i + g_2(x).j + g_3(x).k \,\middle|\, g_0, g_1, g_2, g_3 \in \mathbb{Z}_q[x]/(x^N - 1),\ i^2 = -1, j^2 = -1, ij = -ji = k \right\}.$$

Assume that $q_0, q_1 \in \mathbb{A}_0$ (or $\mathbb{A}_1$), $q_0 = a(x) + b(x).i + c(x).j + d(x).k$ and $q_1 = \alpha(x) + \beta(x).i + \gamma(x).j + \delta(x).k$, then the addition and multiplication of two quaternions, norm and multiplicative inverse are defined in the following way:

- Addition:

$$q_0 + q_1 = (a(x) + \alpha(x)) + (b(x)) + \beta(x).i + (c(x) + \gamma(x)).j + (d(x) + \delta(x)).k.$$

- Multiplication:

$$\begin{aligned}q_0 \times q_1 = &\ (a(x) \star \alpha(x) - b(x) \star \beta(x) - d(x) \star \delta(x) - c(x) \star \gamma(x)) \\ &+ (a(x) \star \beta(x) + b(x) \star \alpha(x) - d(x) \star \gamma(x) + c(x) \star \delta(x)).i \\ &+ (d(x) \star \beta(x) + c(x) \star \alpha(x) + a(x) \star \gamma(x) - b(x) \star \delta(x)).j \\ &+ (b(x) \star \gamma(x) + a(x) \star \delta(x) - c(x) \star \beta(x) + d(x) \star \alpha(x)).k,\end{aligned}$$

  where $\star$ denotes the convolution product.

- Conjugate:

$$\forall q_0 \in \mathbb{A}_0\,(or \mathbb{A}_1) \rightarrow \bar{q}_0 = a(x) - b(x).i - c(x).j - d(x).k.$$

- Norm:

$$\forall q_0 \in \mathbb{A}_0\,(or \mathbb{A}_1) \rightarrow N(q_0) = q_0 \times \bar{q}_0 = a(x)^2 + b(x)^2 + c(x)^2 + d(x)^2.$$

- Multiplicative inverse:

$$\begin{aligned}N(q_0) \neq 0 \rightarrow q_0^{-1} = \tfrac{\bar{q}_0}{N(q_0)} = \left(a(x)^2 + b(x)^2 + c(x)^2 + d(x)^2\right)^{-1}.(a(x) - b(x).i - c(x).j - d(x).k) \\ N(q_1) \neq 0 \rightarrow q_1^{-1} = \tfrac{\bar{q}_1}{N(q_1)} = \left(\alpha(x)^2 + \beta(x)^2 + \gamma(x)^2 + \delta(x)^2\right)^{-1}.(\alpha(x) - \beta(x).i - \gamma(x).j - \delta(x).k).\end{aligned}$$

Note that multiplication of two polynomials and inverse of a polynomial are taken over the underlying ring. For instance, assume that we want to calculate the multiplicative inverse of $q_0 \in \left(\frac{-1,-1}{Z_3[x]/(x^{11}-1)}\right)$ with the following value:

$$\begin{aligned}q0 = (-1 + x + x^2 - x^4 + x^6 + x^9 - x^{10}) + (1 - x + x^3 - x^5 + x^7 + x^8 - x^{10}).i \\ +(-1 + x^2 + x^3 - x^4 + x^5 + x^6 - x^7).j + (-1 + x^2 + x^3 - x^4 + x^6 + x^8 - x^9).k.\end{aligned}$$

We first calculate the norm of q_0 over $\mathbb{Z}_3[x]/(x^{11} - 1)$ as follows:

$$\begin{aligned}N(q0) &= (-1 + x + x^2 - x^4 + x^6 + x^9 - x^{10})^2 + (1 - x + x^3 - x^5 + x^7 + x^8 - x^{10})^2 \\ &+(-1 + x^2 + x^3 - x^4 + x^5 + x^6 - x^7)^2 + (-1 + x^2 + x^3 - x^4 + x^6 + x^8 - x^9)^2 \\ &= (x - x^2 - x^4 + x^5 + x^6 + x^7 - x^8) \text{ (over } \mathbb{Z}_3[x]/(x^{11} - 1)).\end{aligned}$$

By definition, the inverse of q_0 is computed as $\frac{\bar{q}_0}{N(q_0)} = N(q_0)^{-1}.\bar{q}_0$, and

$N(q_0)^{-1} = -x + x^2 + x^3 - x^4 - x^5 + x^6 - x^7 - x^8 - x^9 + x^{10}(\text{over } \mathbb{Z}_3[\text{x}]/(\text{x}^{11} - 1))$

$q_0^{-1} = -1 + x - x^2 + x^4 + x^5 - x^6 - x^7 - x^8 - (1 - x - x^3 + x^4 + x^5 - x^6 + x^7 + x^8 + x^9 + x^{10}).i$
$-(x^2 - x^3 - x^4 - x^6 + x^7 - x^8 + x^9 - x^{10}).j - (1 + x + x^3 + x^4 - x^5 + x^6 - x^7 - x^8 - x^9).k.$

The following operations will be needed for calculation of inverse of an element in $\mathbb{A}_1 = \left(\frac{-1,-1}{\mathbb{Z}_q[x]/(x^N-1)}\right)$:

A Calculation of $g(x) \leftarrow a(x)^2 + b(x)^2 + c(x)^2 + d(x)^2$ over the ring $\left(\frac{-1,-1}{\mathbb{Z}_q[x]/(x^N-1)}\right)$ (including $4.N^2$ multiplications and $3.N$ additions) with the worst-case complexity of $\mathcal{O}(N^2)$.

B Calculation of $g(x)^{-1}$ over the ring $\left(\frac{-1,-1}{\mathbb{Z}_q[x]/(x^N-1)}\right)$ with complexity of $\mathcal{O}(N^2 log(p^2))$.

C Calculation of $g(x)^{-1}.(a(x)-b(x).i-c(x).j-d(x).k)$ (including $4N^2$ multiplications) with the worst-case complexity of $\mathcal{O}(N^2)$.

One can easily prove that the rings $\left(\frac{-1,-1}{\mathbb{Z}_p[x]/(x^N-1)}\right)$ and $\left(\frac{-1,-1}{\mathbb{Z}_q[x]/(x^N-1)}\right)$ are isomorphic to the ring of circulant matrices of dimension $N\times N$ with entries from $\mathbb{F}=\mathbb{Z}_p$ and $\mathbb{F}=\mathbb{Z}_q$, respectively. Consider a vector $v=[v_0,v_1,...,v_{N-1}]\in\mathbb{F}^N$ and define:

$$Circ(v)\doteq\begin{bmatrix} v_0 & v_{N-1} & v_{N-2} & \cdots & \cdots & v_2 & v_1 \\ v_1 & v_0 & v_{N-1} & \ddots & & v_3 & v_2 \\ v_2 & v_1 & v_0 & \ddots & & \vdots & \vdots \\ v_3 & v_2 & v_1 & \ddots & & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & & v_{N-1} & v_{N-2} \\ v_{N-2} & v_{N-3} & \vdots & \ddots & & v_0 & v_{N-1} \\ v_{N-1} & v_{N-2} & \cdots & \cdots & \cdots & v_1 & v_0 \end{bmatrix}.$$

If we represent each polynomial $f(x)$ as a vector of coefficients $f=[f_0,f_1,...,f_{N-1}]$, then the isomorphic representation for elements of $\mathbb{A}_0$ and $\mathbb{A}_1$ will be as follows:
$\mathbb{A}_0\doteq\left(\frac{-1,-1}{\mathbb{Z}_p[x]/(x^N-1)}\right)=\{C_0+C_1.i+C_2.j+C_3.k|C_0,C_1,C_2,C_3\in$ Circulant Matrice of Dimension N over $\mathbb{Z}_p,\ i^2=-1,j^2=-1,ij=-ji=k\}$, and
$\mathbb{A}_1\doteq\left(\frac{-1,-1}{\mathbb{Z}_q[x]/(x^N-1)}\right)=\{C_0+C_1.i+C_2.j+C_3.k|C_0,C_1,C_2,C_3\in$ Circulant Matrice of Dimension N over $\mathbb{Z}_q,\ i^2=-1,j^2=-1,ij=-ji=k\}$.

Therefore, each of the isomorphic representations of $\mathbb{A}_0$ and $\mathbb{A}_1$ can be utilized without any ambiguity. Hence, we will use polynomial representation for the description of the proposed scheme and matrix representation for lattice analysis.

In the matrix representations of $\mathbb{A}_0$ and $\mathbb{A}_1$, an element like $Q\doteq C_0+C_1.i+C_2.j+C_3.k$ can be shown as the following quaternionic matrix in which $Q_{i,j}\doteq C_{0_{(i,j)}}+C_{1_{(i,j)}}.i+C_{2_{(i,j)}}.j+C_{3_{(i,j)}}.k$:

$$Q\doteq\begin{bmatrix} q_0 & q_{N-1} & q_{N-2} & \cdots & \cdots & q_2 & q_1 \\ q_1 & q_0 & q_{N-1} & \ddots & & q_3 & q_2 \\ q_2 & q_1 & q_0 & \ddots & & \vdots & \vdots \\ q_3 & q_2 & q_1 & \ddots & & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & & q_{N-1} & q_{N-2} \\ q_{N-2} & q_{N-3} & \vdots & \ddots & & q_0 & q_{N-1} \\ q_{N-1} & q_{N-2} & \cdots & \cdots & \cdots & q_1 & q_0 \end{bmatrix}.$$

A lot of research has been performed on quaternionic matrices and it seems that they lack many properties that matrices over an arbitrary field $\mathbb{F}$ possess. In particular, the determinant function of quaternionic matrices is not well-defined in general. They also have different left and right eigenvalues and eigenvectors. On the other hand, the existence of inverse for a quaternionic matrix has been proved and can be calculated by a method similar to Gaussian elimination [16, 17]. Consequently, in lattice analysis, many of these relations will lose their applicability and this is highly valuable with regards to the strength of our proposed scheme, QTRU, against lattice attacks.

5 Proposed Scheme: QTRU

Similar to NTRU, the security of the QTRU cryptosystem depends on three parameters (N, p, q) and four subsets $\mathcal{L}_f$, L_m, $\mathcal{L}_\phi$, $\mathcal{L}_g \subset \mathbb{A}$ ($\mathbb{A} \doteq \left(\frac{-1,-1}{\mathbb{Z}[x]/(x^N-1)}\right)$). Here, N, p and q are constant parameters which play a role similar to the equivalent parameters in NTRU. The constants d_f, d_g, d_ϕ , and d_m and the subsets $\mathcal{L}_f$, $\mathcal{L}_\phi$, $\mathcal{L}_g$ and $\mathcal{L}_m$ are defined exactly as in Table (1). Since encryption and decryption are taken place in a multi-dimensional vector space, the following notations and symbols are required:

$$\vec{F} = f_0 + f_1.i + f_2.j + f_3.k \in \left(\frac{-1,-1}{\mathbb{Z}[x]/(x^N-1)}\right), \text{ and}$$

$$\left\{f_0 \triangleq f_0(x), f_1 \triangleq f_1(x), f_2 \triangleq f_2(x), f_3 \triangleq f_3(x)\right\} \in \mathbb{Z}[x]/(x^N - 1).$$

The symbol $\circ$ denotes the quaternionic multiplication and is defined as follows:

$$\begin{aligned}
&\vec{F} \circ \vec{G} = (f_0 + f_1.i + f_2.j + f_3.k) \circ (g_0 + g_1.i + g_2.j + g_3.k) \\
&= \quad (f_0 \star g_0 - f_1 \star g_1 - f_3 \star g_3 - f_2 \star g_2) \\
&\ + (f_0 \star g_1 + f_1 \star g_0 - f_3 \star g_2 + f_2 \star g_3).i \\
&\ +(f_3 \star g_1 + f_2 \star g_0 + f_0 \star g_2 - f_1 \star g_3).j \\
&\ +(f_1 \star g_2 + f_0 \star g_3 - f_2 \star g_1 + f_3 \star g_0).k,
\end{aligned}$$

where $\star$ denotes the convolution product. We denote the conjugate of a quaternion $\vec{F}$ by $\vec{F}^*$. QTRU can now be described as follows:

Key Generation In order to generate a pair of public and private keys, two small quaternion $\vec{F}$ and $\vec{G}$ are randomly generated:

$$\begin{aligned}
\vec{F} &= f_0 + f_1.\mathbf{i} + f_2.\mathbf{j} + f_3.\mathbf{k}, \textit{such that} \;\; f_0, f_1, f_2, f_3 \in \mathcal{L}_f, \\
\vec{G} &= g_0 + g_1.\mathbf{i} + g_2.\mathbf{j} + g_3.\mathbf{k}, \textit{such that} \;\; g_0, g_1, g_2, g_3 \in \mathcal{L}_g.
\end{aligned}$$

The quaternion $\vec{F}$ must be invertible over $\mathbb{A}_0 = \left(\frac{-1,-1}{\mathbb{Z}_p[x]/(x^N-1)}\right)$ and $\mathbb{A}_1 = \left(\frac{-1,-1}{\mathbb{Z}_q[x]/(x^N-1)}\right)$. As mentioned in the previous section, the necessary and sufficient condition for $\vec{F}$ to be invertible over $\mathbb{A}_0$ and $\mathbb{A}_1$ is that the polynomial $\left\|\vec{F}\right\| = (f_0^2 + f_1^2 + f_2^2 + f_3^2)$ be invertible over the rings $\mathbb{Z}_p[x]/(x^N - 1)$ and $\mathbb{Z}_q[x]/(x^N - 1)$. Given the fact that invertiblilty of quaternion $\vec{F}$ depends on the four polynomials f_0, f_1, f_2, f_3, there is more freedom in selection of these polynomials. For example, there is no necessity for selecting all the polynomials from $\mathcal{L}_f$, as it is sufficient to have

$f_0^2+f_1^2+f_2^2+f_3^2\big|_{x=1} \neq 0$ ($mod\ p$ and q). If the generated quaternion is not invertible over $\mathbb{A}_0$ and $\mathbb{A}_1$, a new quaternion can easily be generated.

After generation of $\vec{F}$ and $\vec{G}$, the inverses of $\vec{F}$ (denoted by $\vec{F}_p$ and $\vec{F}_q$) will be computed in the following way:

$$\vec{F}_p = \big\langle (f_0^2+f_1^2+f_2^2+f_3^2)^{-1}\ over\ \mathbb{Z}_p[x]/(x^N-1)\big\rangle \circ \vec{F}^* = \mu_0+\mu_1.i+\mu_2.j+\mu_3.k,$$

$$\vec{F}_q = \big\langle (f_0^2+f_1^2+f_2^2+f_3^2)^{-1}\ over\ \mathbb{Z}_q[x]/(x^N-1)\big\rangle \circ \vec{F}^* = \eta_0+\eta_1.i+\eta_2.j+\eta_3.k.$$

Now, the public-key, which is a quaternion, is calculated and then made public as follows:

$$\begin{aligned}
&\vec{H} = \vec{F}_q \circ \vec{G} = \\
&(\eta_0 \star g_0 - \eta_1 \star g_1 - \eta_3 \star g_3 - \eta_2 \star g_2)+ \\
&(\eta_0 \star g_1 + \eta_1 \star g_0 - \eta_3 \star g_2 + \eta_2 \star g_3).i+ \\
&(\eta_3 \star g_1 + \eta_2 \star g_0 + \eta_0 \star g_2 - \eta_1 \star g_3).j+ \\
&(\eta_1 \star g_2 + \eta_0 \star g_3 - \eta_2 \star g_1 + \eta_3 \star g_0).k.
\end{aligned}$$

The quaternions $\vec{F}$, $\vec{F}_p$ and $\vec{F}_q$ will be kept secret in order to be used in the decryption phase. One can estimate that the key generation of QTRU is 16 times slower than that of NTRU, when the same parameters (N, p, q) are used in both cryptosystems. However, in QTRU, we can work with a lower dimension, without reducing the system security.

We note that if coefficients of i, j, and k are all zero in the quaternions $\vec{F}$ and $\vec{G}$, then the QTRU cryptosystem will be completely analogous to NTRU. On the other hand, if the coefficients of j and k are equal to zero, a cryptosystem similar to the one proposed in [6] will be acquired.

Encryption In the encryption process, the cryptosystem initially generates a random quaternion, called the blinding quaternion. Incoming data must be converted into a quaternion including four small polynomials. Data conversion into polynomials is performed exactly similar to the NTRU system. However, we have much more freedom for formation of a quaternion with four elements. The incoming data can be generated from the same or four different sources but transformed into one quaternion based on a simple conversion. After the conversion of the incoming message(s) into one quaternion, the ciphertext will be computed and sent in the following way:
Data Quaternion:

$$\vec{M} = m_0 + m_1.\mathbf{i} + m_2.\mathbf{j} + m_3.\mathbf{k},$$
$$m_0 \triangleq m_0(x), m_1 \triangleq m_1(x), m_2 \triangleq m_2(x), m_3 \triangleq m_3(x) \in \mathcal{L}_m.$$

Blinding Quaternion :

$$\vec{\Phi} = \phi_0 + \phi_1.\mathbf{i} + \phi_2.\mathbf{j} + \phi_3.\mathbf{k},$$
$$\phi_0 \triangleq \phi_0(x), \phi_1 \triangleq \phi_1(x), \phi_2 \triangleq \phi_2(x), \phi_3 \triangleq \phi_3(x) \in \mathcal{L}_\phi.$$

Ciphertext:

$$\vec{E} = p.\vec{H} \circ \vec{\Phi} + \vec{M}.$$

Encryption needs one quaternionic multiplication including 16 convolution multiplications with $\mathcal{O}(N^2)$ complexity, and 4 polynomial additions with $\mathcal{O}(N)$ complexity. In the encryption phase, a total of four data vectors are encrypted at once.

Decryption In order to decrypt, the received quaternion $\vec{E}$ is multiplied by the private key $\vec{F}$:

$$\begin{aligned} \vec{F} \circ \vec{E} &= (\vec{F} \circ (p.\vec{H} \circ \vec{\Phi} + \vec{M}) \mod q \\ &= (\vec{F} \circ p.\vec{H} \circ \vec{\Phi} + \vec{F} \circ \vec{M}) \mod q \\ &= (p.\vec{F} \circ \vec{F}_q \circ \vec{G} \circ \vec{\Phi} + \vec{F} \circ \vec{M}) \mod q \\ &= (p.\vec{G} \circ \vec{\Phi} + \vec{F} \circ \vec{M}). \end{aligned}$$

The coefficients of the four polynomials in the resulting quaternion must be reduced mod q into the interval $(-q/2, +q/2]$. Upon suitable selection of the cryptosystem constant parameters, the coefficients of the polynomial $(p.\vec{G} \circ \vec{\Phi} + \vec{F} \circ \vec{M})$ will most probably be within $(-q/2, +q/2]$ and the last reduction mod q will not be required. With such an assumption, when the result of $(p.\vec{G} \circ \vec{\Phi} + \vec{F} \circ \vec{M})$ is reduced mod p, the term $p.\vec{G} \circ \vec{\Phi}$ will be removed and the $\vec{F} \circ \vec{M} \pmod{p}$ will remain. In order to extract the original message $\vec{M}$, it will suffice to multiply $\vec{F} \circ \vec{M} \pmod{p}$ by $\vec{F}_p$ and adjust the resulting coefficients within the interval $[-p/2, +p/2]$. Therefore, decryption includes 32 convolutions and naturally decryption speed becomes half the encryption speed, analogous to the NTRU cryptosystem.

6 Analyzing QTRU

In this section, we analyze QTRU and discuss the probability of successful decryption, key security, message security and the message expansion rate. Moreover, we suggest a set of parameters for the QTRU cryptosystem.

Successful Decryption Probability of successful decryption in the QTRU is calculated in the same way as NTRU and under the same assumptions considered in [5] and [6]. Moreover, for successful decryption in QTRU, all quaternion coefficients of $\vec{F} \circ \vec{E} = (p.\vec{G} \circ \vec{\Phi} + \vec{F} \circ \vec{M})$ must lie in the interval $\left[\frac{-q+1}{2}, \frac{+q-1}{2}\right]$. Hence, we obtain

$$\vec{A} := \vec{F} \circ \vec{E} = (p.\vec{G} \circ \vec{\Phi} + \vec{F} \circ \vec{M}) = a_0 + a_1.i + a_2.j + a_3.k,$$

where:

$$a_0 = p.g_0 \star \phi_0 - p.g_1 \star \phi_1 - p.g_3 \star \phi_3 - p.g_2 \star \phi_2 + f_0 \star m_0 - f_1 \star m_1 - f_3 \star m_3 - f_2 \star m_2 \doteq [a_{0,0}, a_{0,1}, ..., a_{0,N-1}],$$

$$a_1 = p.g_0 \star \phi_1 + p.g_1 \star \phi_0 - p.g_3 \star \phi_2 + p.g_2 \star \phi_3 + f_0 \star m_1 + f_1 \star m_0 - f_3 \star m_2 + f_2 \star m_3 \doteq [a_{1,0}, a_{1,1}, ..., a_{1,N-1}],$$

$$a_2 = p.g_3 \star \phi_1 + p.g_2 \star \phi_0 + p.g_0 \star \phi_2 - p.g_1 \star \phi_3 + f_3 \star m_1 + f_2 \star m_0 + f_0 \star m_2 - f_1 \star m_3 \doteq [a_{2,0}, a_{2,1}, ..., a_{2,N-1}],$$

$$a_3 = p.g_1 \star \phi_2 + p.g_0 \star \phi_3 - p.g_2 \star \phi_1 + p.g_3 \star \phi_0 + f_1 \star m_2 + f_0 \star m_3 - f_2 \star m_1 + f_3 \star m_0 \doteq [a_{3,0}, a_{3,1}, ..., a_{3,N-1}].$$

Before calculating the probability for successful decryption, one can easily estimate with a glance at the above relations that if we consider all NTRU assumptions, the expected value for all coefficients of a_0, a_1, a_2, a_3 remain equal to zero and their variance quadruples. We know that $f_i \star m_j (i, j = 0, 1, 2, 3)$ and $g_i \star \phi_j (i, j = 0, 1, 2, 3)$ are the products of two small polynomials and that the coefficients of f_i, g_i, and ϕ_i are assumed to be independent random variables that randomly take one of the values: -1, 0, and +1. Now, according to definition of the subsets $\mathcal{L}_f$ and $\mathcal{L}_g$ from Table 1, we obtain:

$$f_i = [f_{i,0}, f_{i,1}, ..., f_{i,N-1}] \qquad i = 0, 1, 2, 3,$$

$$g_i = [g_{i,0}, g_{i,1}, ..., g_{i,N-1}] \qquad i = 0, 1, 2, 3,$$

$$\phi_i = [\phi_{i,0}, \phi_{i,1}, ..., \phi_{i,N-1}] \qquad i = 0, 1, 2, 3,$$

$$\Pr(f_{i,j} = 1) = \frac{d_f}{N}, \qquad \Pr(f_{i,j} = -1) = \frac{d_f - 1}{N} \approx \frac{d_f}{N}, \qquad \Pr(f_{i,j} = 0) = \frac{N - 2d_f}{N} [0, 1],$$

$$\Pr(g_{i,j} = 1) = \Pr(g_{i,j} = -1) = \frac{d_g}{N}, \qquad \Pr(g_{i,j} = 0) = \frac{N - 2d_g}{N},$$

$$\Pr(\phi_{i,j} = 1) = \Pr(\phi_{i,j} = -1) = \frac{d_\phi}{N}, \qquad \Pr(\phi_{i,j} = 0) = \frac{N - 2d_\phi}{N},$$

$$\Pr(m_{i,j} = j) = \frac{1}{p}, \qquad i = 0, 1, 2, 3 \qquad j = \frac{-p+1}{2} ... \frac{+p-1}{2}.$$

Under the above assumptions, we get: $E[f_{i,j}] \approx 0$, $E[g_{i,j}] = 0$, $E[r_{i,j}] = 0$, and $E[m_{i,j}] = 0$. Therefore, we have:

$$E[a_{i,j}] = 0 \qquad i = 0, 1, 2, 3 \qquad j = 0, ..., N-1.$$

In order to calculate $Var[a_{i,j}]$, analogous to NTRU, it is sufficient to write:

$$Var[\phi_{i,k}.g_{j,l}] = \frac{4d_\phi.d_g}{N^2} \qquad i, j = 0, 1, 2, 3 \qquad k, l = 0, ..., N-1,$$

$$Var[f_{i,k}.m_{j,l}] = \frac{d_f(p-1).(p+1)}{6.N} \qquad i, j = 0, 1, 2, 3 \qquad k, l = 0, ..., N-1.$$

As a result,

$$Var[a_{0,k}] = Var[\sum_{\substack{i+j=k \\ (\bmod N)}} (p.g_{0,i} \star \phi_{0,j} - p.g_{1,i} \star \phi_{1,j} - p.g_{3,i} \star \phi_{3,j} - p.g_{2,i} \star \phi_{2,j} + f_{0,i} \star m_{0,j} - f_{1,i} \star m_{1,j} - f_{3,i} \star m_{3,j} - f_{2,i} \star m_{2,j}$$

Upon insertion of $Var[\phi_{i,k} g_{j,l}]$ and $Var[f_{i,k} m_{j,l}]$ values, we obtain:

$$Var[a_{0,k}] = \frac{16p^2 d_\phi d_g}{N} + \frac{4d_f(p-1)(p+1)}{6}.$$

Similarly, we have:

$$Var[a_{1,k}] = Var[a_{2,k}] = Var[a_{3,k}] = \frac{16p^2 d_\phi d_g}{N} + \frac{4d_f(p-1)(p+1)}{6}.$$

It is desirable to calculate the probability that $a_{i,k}$ lies within $\frac{-q+1}{2} ... \frac{+q-1}{2}$, which implies successful decryption. With the assumption that $a_{i,k}$'s have normal distribution with zero mean and the variance calculated as above, we have:

$$\Pr\left(|a_{i,k}| \leq \frac{q-1}{2}\right) = \Pr\left(-\frac{q-1}{2} \leq a_{i,k} \leq \frac{q-1}{2}\right)$$

$$= 2\Phi\left(\frac{q-1}{2\sigma}\right) - 1,$$

where Φ denotes the distribution of the standard normal variable and $\sigma = \sqrt{\frac{16p^2 d_\phi d_g}{N} + \frac{4d_f(p-1)(p+1)}{6}}$.

Regarding the assumption that $a_{i,k}$'s are independent random variables, the probability for successful decryption in QTRU can be calculated through the following two observations:

- The probability for each of the messages m_0, m_1, m_2, or m_3 to be correctly decrypted is:

$$\left(2\Phi(\frac{q-1}{2\sigma}) - 1\right)^N.$$

- The probability for all the messages m_0, m_1, m_2, and m_3 to be correctly decrypted is:

$$\left(2\Phi(\frac{q-1}{2\sigma}) - 1\right)^{4.N}.$$

It is apparent that in QTRU, the variance of the coefficients $(p.\vec{G} \circ \vec{\Phi} + \vec{F} \circ \vec{M})$ increases by a factor of 4 and hence the probability for decryption failure increases. In return, constant parameters of the system, including d_ϕ , d_g, d_f, q, and N, can be chosen in such a way that the decryption failure rate in the QTRU remains equal to that of NTRU. The rightmost column of Table (2) shows the probability for successful decryption for some proposed values of d_ϕ , d_g, d_f, q, and N.

Brute Force Attack In QTRU, an attacker knows the constant and public parameters, namely d_ϕ , d_g, d_f, q, and N, as well as the public-key $\vec{H} = \vec{F}_q \circ \vec{G} = h_0 + h_1.i + h_2.j + h_3.k$. If the attacker can access one of the quaternions $\vec{G} \in \mathcal{L}_g$ or $\vec{F} \in \mathcal{L}_f$, the private key will be easily revealed. In order to find $\vec{G}$ or $\vec{F}$, using a brute force attack, the attacker can try all possible values and check to see if $\vec{F} \circ \vec{H}$ $(\vec{G} \circ \vec{H}^{-1})$ turns into a quaternion with small coefficients or not. The total state space for the two subsets $\mathcal{L}_f$ and $\mathcal{L}_g$ is calculated as follows:

$$|\mathcal{L}_f| = \begin{pmatrix} N \\ d_f \end{pmatrix}^4 \begin{pmatrix} N - d_f + 1 \\ d_f \end{pmatrix}^4 = \frac{(N!)^4}{(d_f!)^8 (N-2d_f)!^4},$$

Pr(Successful Decryption)	*Message Expansion*	*Message Security*	*Key Security*	d_f	d_g	d_f	q	p	N	*Security Level*
0.9997119974	≈4.4	7.8404×10^{31}	1.8356×10^{60}	5	12	15	127	3	107	Moderate
0.9999971752	≈4.4	1.9527×10^{53}	1.8356×10^{60}	10	12	20	191	3	107	
0.9999998808	≈4.6	3.3811×10^{59}	7.7751×10^{67}	10	12	20	191	3	149	High
0.9999845041	≈4.6	7.7751×10^{67}	1.5965×10^{79}	12	15	22	191	3	149	
0.9999563737	≈4.6	1.5965×10^{79}	1.9861×10^{95}	15	20	50	255	3	149	
0.9994484943	≈4.6	1.9864×10^{95}	2.8775×10^{108}	20	25	35	255	3	149	
0.9999808954	≈4.7	1.8749×10^{93}	7.111×10^{99}	18	20	40	255	3	167	
0.9999167707	≈4.7	1.8749×10^{93}	7.111×10^{99}	18	20	50	255	3	167	
0.9993964435	≈4.7	9.60×10^{105}	4.91×10^{111}	22	24	40	255	3	167	
0.9999974680	≈4.8	2.34×10^{101}	9.24×10^{108}	18	20	40	255	3	211	Highest
0.9999782250	≈4.8	1.38×10^{116}	8.37×10^{122}	22	24	30	255	3	211	
0.9999995888	≈5.1	1.13×10^{108}	2.93×10^{116}	18	20	40	255	3	257	
0.9999923928	≈5.1	1.29×10^{132}	1.29×10^{132}	24	24	30	255	3	257	

Table 2: the probability of successful encryption in the QTRU, security level of the private key, and message security according to some generic parameters d_ϕ , d_g, d_f, p, q, N.

$$|\mathcal{L}_g| = \begin{pmatrix} N \\ d_g \end{pmatrix}^4 \begin{pmatrix} N - d_g + 1 \\ d_g \end{pmatrix}^4 = \frac{(N!)^4}{(d_g!)^8 (N - 2d_g)!^4}.$$

Since d_g is generally considered to be smaller than d_f, evidently $\mathcal{L}_g$ is smaller than $\mathcal{L}_f$ and by trying all possible values of $\vec{G} \in \mathcal{L}_g$ in $\vec{G} \circ \vec{H}^{-1}$, the attacker can reveal the private key through searching a space of order $\mathcal{L}_g$ and with resort to Meet-In-The-Middle attack through searching a space of order $\sqrt{|\mathcal{L}_g|} = \frac{(N!)^2}{(d_g!)^4 (N-2d_g)!^2}$ [18]. Similarly, in order to find the original message from the corresponding ciphertext, the attacker must search in $\mathcal{L}_\phi$. On average, the search must be done in a space of order $\sqrt{|\mathcal{L}_\phi|} = \frac{(N!)^2}{(d_\phi!)^4 (N-2d_\phi)!^2}$. However, with the typical values for d_ϕ , d_g, and N, finding the private key or plaintext using brute force attack is computationally infeasible.

In Table (2), *Key Security* and *Message Security* columns respectively indicate the search space for the private key ($\sqrt{|\mathcal{L}_g|}$) and the search space for the message ($\sqrt{|\mathcal{L}_g|}$) with regards to typical values of d_ϕ , d_g, and N. Therefore, the QTRU cryptosystem seems to be completely secure against brute force attack. Moreover, with regards to chosen plaintext-attack, all analyses and solutions proposed for NTRU, work just as well for the QTRU.

Message Expansion Analogous to NTRU, the length of the encrypted message in QTRU is more than the original message and that is part of the price one pays for gaining more speed in both systems. The expansion ratio can be easily calculated through $\frac{\log |C|}{\log |P|} = \frac{\log q^{4N}}{\log p^{4N}} = \frac{\log q}{\log p}$, where C is the state space for the encrypted message and P is the state space for plaintext; for NTRU and the QTRU, this ratio depends merely on p and q. Table (2) presents the message expansion rate for some typical values of p and q. Message expansion rate for generic parameters in both NTRU and QTRU fluctuates between 4 and 5. In recent years, NTRU has been thoroughly analyzed and its resistance against lattice attacks has been sufficiently studied implying its main core to be secure, see for example [7], [3], [10], [9], and [5].

7 Analyzing Lattice Attacks Against QTRU

Given the fact that quaternion algebra is a non-commutative algebraic structure, it is implies that lattice-based attacks against QTRU are generally more difficult. This is because lattice theory inherently relies on the commutativity in the commutative rings while quaternionic matrices or lattices inherently possess certain complexities which do not seem to be solvable [21]. For the sake of clarity, we divide the analysis of lattice-based attacks against QTRU into two parts: *Partial Lattice Attack* and *Full Lattice Attack*.

Partial Lattice Attack Let each quaternion isomorphic representation on $\mathcal{R}^4$ be considered as follows:

$$q \triangleq q_0 + q_1 i + q_2 j + q_3 k \cong \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \\ -q_1 & q_0 & -q_3 & q_2 \\ -q_2 & q_3 & q_0 & -q_1 \\ -q_3 & -q_2 & q_1 & q_0 \end{bmatrix},$$

$$x \triangleq x_0 + x_1 i + x_2 j + x_3 k \cong \begin{bmatrix} x_0 & x_1 & x_2 & x_3 \\ -x_1 & x_0 & -x_3 & x_2 \\ -x_2 & x_3 & x_0 & -x_1 \\ -x_3 & -x_2 & x_1 & x_0 \end{bmatrix},$$

$$x \circ q = \begin{bmatrix} x_0 & x_1 & x_2 & x_3 \\ -x_1 & x_0 & -x_3 & x_2 \\ -x_2 & x_3 & x_0 & -x_1 \\ -x_3 & -x_2 & x_1 & x_0 \end{bmatrix} \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \\ -q_1 & q_0 & -q_3 & q_2 \\ -q_2 & q_3 & q_0 & -q_1 \\ -q_3 & -q_2 & q_1 & q_0 \end{bmatrix},$$

where $\circ$ denotes quaternionic multiplication.

The attacker knows the constant parameters of the system d_ϕ , d_g, d_f, p, q, and N, as well as the public-key $\vec{H} = \vec{F_q} \circ \vec{G} = h_0 + h_1.i + h_2.j + h_3.k$. Obviously, once the attacker manages to find one of the quaternion $\vec{F}$ or $\vec{G}$, the QTRU cryptosystem breaks. It is known that h_0, h_1, h_2, and h_3, in the public-key $\vec{H} = h_0 + h_1 i + h_2 j + h_3 k$, are polynomials over $\mathbb{Z}[x]/(x^N - 1)$. We also represent those polynomials in their isomorphic representation as vectors over $\mathbb{Z}^N$ as:

$$\begin{aligned}
&\vec{H} = h_0 + h_1 i + h_2 j + h_3 k \triangleq \begin{bmatrix} h_0 & h_1 & h_2 & h_3 \end{bmatrix}, \\
&h_0 = h_{0,0} + h_{0,1}.x + \ldots + h_{0,N-2}.x^{N-2} + h_{0,N-1}.x^{N-1} \triangleq \begin{bmatrix} h_{0,0} & h_{0,1} & h_{0,2} & \ldots & h_{0,N-2} & h_{0,N-1} \end{bmatrix} \in \mathbb{Z}^N, \\
&h_1 = h_{1,0} + h_{1,1}.x + \ldots + h_{1,N-2}.x^{N-2} + h_{1,N-1}.x^{N-1} \triangleq \begin{bmatrix} h_{1,0} & h_{1,1} & h_{1,2} & \ldots & h_{1,N-2} & h_{1,N-1} \end{bmatrix} \in \mathbb{Z}^N, \\
&h_2 = h_{2,0} + h_{2,1}.x + \ldots + h_{2,N-2}.x^{N-2} + h_{2,N-1}.x^{N-1} \triangleq \begin{bmatrix} h_{2,0} & h_{2,1} & h_{2,2} & \ldots & h_{2,N-2} & h_{2,N-1} \end{bmatrix} \in \mathbb{Z}^N, \\
&h_3 = h_{3,0} + h_{3,1}.x + \ldots + h_{3,N-2}.x^{N-2} + h_{3,N-1}.x^{N-1} \triangleq \begin{bmatrix} h_{3,0} & h_{3,1} & h_{3,2} & \ldots & h_{3,N-2} & h_{3,N-1} \end{bmatrix} \in \mathbb{Z}^N.
\end{aligned}$$

As indicated in Section 4, $\mathbb{Z}[x]/(x^N - 1)$ is isomorphic to the circulant matrices ring of order N over $\mathbb{Z}$. Hence, we also represent h_0, h_1, h_2, and h_3 polynomials in their isomorphic representation for lattice analysis:

$$(h_i)_{N\times N} \triangleq \begin{bmatrix} h_{i,0} & h_{i,1} & h_{i,2} & \cdots & h_{i,N-1} \\ h_{i,N-1} & h_{i,0} & h_{i,1} & & h_{i,N-2} \\ h_{i,N-2} & h_{i,N-1} & h_{i,0} & & h_{i,N-3} \\ \vdots & & & \ddots & \vdots \\ & & & & \\ h_{i,2} & h_{i,3} & & & \\ h_{i,1} & h_{i,2} & & \cdots & h_{i,0} \end{bmatrix} \qquad i = 0, 1, 2, 3.$$

Under the above assumptions, a partial lattice attack can be described as follows. Let's denotes the quaternion $\vec{F}$ and $\vec{G}$ as $\vec{F} \triangleq \begin{bmatrix} f_0 & f_1 & f_2 & f_3 \end{bmatrix}$ and $\vec{G} \triangleq \begin{bmatrix} g_0 & g_1 & g_2 & g_3 \end{bmatrix}$, where f_0, f_1, f_2, f_3 and $g_0, g_1, g_2, g_3 \in \mathbb{Z}[x]/(x^N - 1)$. Then, it is clear that the collection of pairs of vectors $[u_0, u_1, u_2, u_3, v_0, v_1, v_2, v_3] \in \mathbb{Z}^{8N}$ satisfying the congruence $\vec{F} \circ \vec{H} = \vec{G}$ form a lattice in $\mathbb{Z}^{8N}$. This lattice which has been denoted as $\mathcal{L}_{Partial}$ is defined as follows:

$$\mathcal{L}_{Partial} = RowSpan \left[\begin{array}{c|cccc} \lambda.I_{4N\times 4N} & (h_0)_{N\times N} & (h_1)_{N\times N} & (h_2)_{N\times N} & (h_3)_{N\times N} \\ & (-h_1)_{N\times N} & (h_0)_{N\times N} & (-h_3)_{N\times N} & (h_2)_{N\times N} \\ & (-h_2)_{N\times N} & (h_3)_{N\times N} & (h_0)_{N\times N} & (-h_1)_{N\times N} \\ & (-h_3)_{N\times N} & (-h_2)_{N\times N} & (h_1)_{N\times N} & (h_0)_{N\times N} \\ \hline 0_{4N\times 4N} & \multicolumn{4}{c}{q.I_{4N\times 4N}} \end{array}\right] \in \mathbb{Z}^{4N\times 4N}.$$

The lattice $\mathcal{L}_{Partial}$ includes all vectors in the form of $[u_0, u_1, u_2, u_3, v_0, v_1, v_2, v_3] \in \mathbb{Z}^{8N}$, which satisfy $\vec{F} \circ \vec{H} = \vec{G}$. However, the fundamental difference between the NTRU and QTRU lattice is that all points spanned by this lattice merely encompass a partial subset of the total set of vectors which satisfy $\vec{F} \circ \vec{H} = \vec{G}$. To see this, let $[u_0, u_1, u_2, u_3, v_0, v_1, v_2, v_3]$ be a vector satisfying $\vec{F} \circ \vec{H} = \vec{G}$. Then, $[-u_1, u_0, -u_3, u_2, -v_1, v_0, -v_3, v_2]$, too, will be an answer for (since $\vec{F} \circ \vec{H} = \vec{G} \rightarrow i.\vec{F} \circ \vec{H} = i.\vec{G}$) but $\mathcal{L}_{Partial}$ will not necessarily include such answers. **000 this part is very unclear, consider rewriting. the propositions are making it more difficult**

If the attacker manages to find a short vector in this lattice using a lattice reduction algorithm, s/he is capable of finding the private key because such a short vector will satisfy $\vec{F} \circ \vec{H} = \vec{G}$. However, even with such an optimistic assumption, $\mathcal{L}_{Partial}$ have dimensions which are four times larger than those of NTRU lattice. Note that QTRU deals with $(N = 107, p, q)$ just as NTRU does with $(N = 428, p, q)$. It can be asserted that with any selective (N, p, q), QTRU (that acts approximately four times slower, compared to NTRU) has a security equal to that of NTRU with $(4N, p, q)$ dimensions. While NTRU with $4N$ dimensions is sixteen times slower than NTRU with dimension of N. In more precise terms, QTRU with (N, p, q) dimensions, has a security equal to NTRU with $(4N, p, q)$ and QTRU with (N, p, q) is four times faster than NTRU with $(4N, p, q)$. The main point to emphasize is that with an advantage of smaller dimensions, QTRU can present a higher security than NTRU. **000 I suggest that you add a paragraph like this in the abstract or introduction to show the strength of QTRU early on**

In practice, partial lattice attacks do not always succeed because generally $\mathcal{L}_{Partial}$, and even its variants such as the lattice $i.\vec{F} \circ \vec{H} = i.\vec{G}$, $j.\vec{F} \circ \vec{H} = j.\vec{G}$, and $k.\vec{F} \circ \vec{H} = k.\vec{G}$, do not necessarily include all answers of $\vec{F} \circ \vec{H} = \vec{G}$ in a way that $f_0, f_1, f_2, f_3, g_0, g_1, g_2, g_3$ be short vectors (i.e., polynomials with very small coefficients). Therefore, we must find out what lattices include all vectors satisfying the congruence $\vec{F} \circ \vec{H} = \vec{G}$.

Full Lattice Attack Before looking at the lattice containing all vectors satisfying the congruence $\vec{F} \circ \vec{H} = \vec{G}$, we introduce the *Quaternionic Lattice* first. Suppose $\vec{v}_1$, $\vec{v}_2$, ..., $\vec{v}_N$ are quaternionic vectors in $\mathbb{H}^N$ ($\mathbb{H} = \left(\frac{-1,-1}{\mathbb{R}}\right)$). Moreover, let $\left(\frac{-1,-1}{\mathbb{Z}}\right)$ be a quaternion with integer elements (such quaternions often are called Lipschitz integers). Given non-commutativity of the quaternion algebra, *Left/Right Quaternionic Lattice ($\mathcal{LQL}/\mathcal{RQL}$)* can be defined in the following way:

$$\mathcal{LQL} = \left\{ q_1 \circ \vec{v}_1 + q_2 \circ \vec{v}_2 + q_3 \circ \vec{v}_3 + \cdots + q_N \circ \vec{v}_N | q_i \in \left(\frac{-1,-1}{\mathbb{Z}}\right) \right\},$$

$$\mathcal{RQL} = \left\{ \vec{v}_1 \circ q_1 + \vec{v}_2 \circ q_2 + \vec{v}_3 \circ q_3 + \cdots + \vec{v}_N \circ q_N | q_i \in \left(\frac{-1,-1}{\mathbb{Z}}\right) \right\}.$$

A General Quaternionic Lattice is a quaternionic combinations (versus linear combinations) of vectors $\vec{v}_1$, $\vec{v}_2$, ..., $\vec{v}_N$ from left or right by integer quaternion $q_i \in \left(\frac{-1,-1}{Z}\right)$. Now let's look into the *Quaternionic Lattice* of QTRU: As indicated in Section 4, the public-key $\vec{H}$ can be represented as follows:

$$\vec{H} \doteq H_0 + H_1.i + H_2.j + H_3.k, \qquad H_0 \triangleq Circ(h_0), H_1 \triangleq Circ(h_1), H_2 \triangleq Circ(h_2), H_3 \triangleq Circ(h_3).$$

In other words,

$$\vec{H} \triangleq \begin{bmatrix} h_{0,0} & h_{0,1} & h_{0,2} & \cdots & h_{0,N-1} \\ h_{0,N-1} & h_{0,0} & h_{0,1} & & h_{0,N-2} \\ h_{0,N-2} & h_{0,N-1} & h_{0,0} & & h_{0,N-3} \\ \vdots & & & \ddots & \vdots \\ & & & & \\ h_{0,2} & h_{0,3} & & & \\ h_{0,1} & h_{0,2} & & \cdots & h_{0,0} \end{bmatrix} + \begin{bmatrix} h_{1,0} & h_{1,1} & h_{1,2} & \cdots & h_{1,N-1} \\ h_{1,N-1} & h_{1,0} & h_{1,1} & & h_{1,N-2} \\ h_{1,N-2} & h_{1,N-1} & h_{1,0} & & h_{1,N-3} \\ \vdots & & & \ddots & \vdots \\ & & & & \\ h_{1,2} & h_{1,3} & & & \\ h_{1,1} & h_{1,2} & & \cdots & h_{1,0} \end{bmatrix} .i +$$

$$\begin{bmatrix} h_{1,0} & h_{1,1} & h_{1,2} & \cdots & h_{1,N-1} \\ h_{2,N-1} & h_{2,0} & h_{2,1} & & h_{2,N-2} \\ h_{2,N-2} & h_{2,N-1} & h_{2,0} & & h_{2,N-3} \\ \vdots & & & \ddots & \vdots \\ & & & & \\ h_{2,2} & h_{2,3} & & & \\ h_{2,1} & h_{2,2} & & \cdots & h_{2,0} \end{bmatrix} .j + \begin{bmatrix} h_{1,0} & h_{1,1} & h_{1,2} & \cdots & h_{1,N-1} \\ h_{3,N-1} & h_{3,0} & h_{3,1} & & h_{3,N-2} \\ h_{3,N-2} & h_{3,N-1} & h_{3,0} & & h_{3,N-3} \\ \vdots & & & \ddots & \vdots \\ & & & & \\ h_{3,2} & h_{3,3} & & & \\ h_{3,1} & h_{3,2} & & \cdots & h_{3,0} \end{bmatrix} .k,$$

$$\vec{H}_{N\times N} \triangleq \begin{bmatrix} h_{0,0}+h_{1,0}i+h_{2,0}j+h_{3,0}k & \cdots & h_{0,N-1}+h_{1,N-1}i+h_{2,N-1}j+h_{3,N-1}k \\ h_{0,N-1}+h_{1,N-1}i+h_{2,N-1}j+h_{3,N-1}k & & h_{0,N-2}+h_{1,N-2}i+h_{2,N-2}j+h_{3,N-2}k \\ h_{0,N-2}+h_{1,N-2}i+h_{2,N-2}j+h_{3,N-2}k & & h_{0,N-3}+h_{1,N-3}i+h_{2,N-3}j+h_{3,N-3}k \\ \vdots & \ddots & \vdots \\ & & \\ h_{0,2}+h_{1,2}i+h_{2,2}j+h_{3,2}k & & \\ h_{0,1}+h_{1,1}i+h_{2,1}j+h_{3,1}k & \cdots & h_{0,0}+h_{1,0}i+h_{2,0}j+h_{3,0}k \end{bmatrix}.$$

Given the above definitions, the attacker can form the following quaternionic lattice using the public-key $\vec{H}$:

$$\mathcal{QL}_{\mathcal{QTRU}} = \left[\begin{array}{c|c} I_{N\times N} & \vec{H}_{N\times N} \\ \hline 0_{N\times N} & q.I_{N\times N} \end{array}\right].$$

Note that $\mathcal{QL}_{\mathcal{QTRU}}$ has dimension equal to $2N \times 2N$. Obviously, the quaternionic combinations of the rows of this matrix are answers for the congruence $\vec{F} \circ \vec{H} = \vec{G}$. To be more precise, all of the left quaternionic (vesus linear) combination of rows of $\mathcal{QL}_{\mathcal{QTRU}}$ satisfy $\vec{F} \circ \vec{H} = \vec{G}$. Now, let's discuss how the attacker can use a lattice reduction algorithm in searching for such short vectors (i.e., rows of the matrix with low norm quaternionic coefficients). Here, we face many serious challenges and open questions:

- As mentioned in section 4, quaternionic matrices (as matrices which have been defined over a Skew Field) lack many properties of matrices $M_{N\times N}(\mathbb{F})$ which are generally defined over a field (or commutative ring). Therefore, many numerical and computational methods provided for matrices $M_{N\times N}(\mathbb{F})$ are not going to work with quaternionic matrices.

- Since determinant is not generally well-defined for quaternionic matrices, many basic concepts of a lattice, such as *unimodular matrices* (e.g., matrices with $det(U) = \pm 1$ which have lattice preserving properties), *determinant of a lattice* and *fundamental parallelepiped volume* lose their meanings. Moreover, such useful and effective propositions as Blichfeldt and Minkowski theorem as well as Gaussian Heuristic lose their efficiency with quaternionic lattices. Although research on quaternionic matrices and their determinant has been going on for the past century [19], what we know about them today is that generally determinant, as a multiplicative homomorphism, is not well-defined for quaternionic matrices [16, 17]. According to [20], determinant mapping for quaternionic matrices can be defined in the following way. For quaternionic matrices, the determinant is defined in terms of the cosets modulo the commutator subgroup of the nonzero elements. Elaborate researches on quaternionic lattice are underway. Yet, in most of those researches, quaternionic lattices are analyzed in specific conditions and on the basis of usage. Hence, according to the knowledge at hand, finding short vectors in quaternionic lattices is facing some difficulties that do not seem to be solvable. For instance, even a simple linear equation with a single quaternionic variable x in the form of $c = \sum_{j=1}^{n} a_i x b_j$ where a_i, b_i, c, are quaternions, for $n \geq 3$, is not solvable in general [21]. In the case of confidence in existence of such an answer, one must find the answer by trial and error.

- None of the lattice reduction algorithms, for example LLL (or its faster variants, e.g., BKZ) have been provided for quaternionic lattice and hence they cannot be used to find short vectors in $\mathcal{QL}_{\mathcal{QTRU}}$ with dimensions of $2N \times 2N$.

Hence, we argue that not only finding short vectors in quaternionic lattices is NP-Hard (as the same problem is NP-Hard even in regular lattice [22]), but also lattice reduction algorithms do not help in reducing the search space in a quaternionic lattice. Therefore, the only effective way to attack the QTRU, is finding short vectors in a lattice with $8N \times 8N$ dimensions by *Partial Lattice Attack method*.

8 Conclusion

In this paper, QTRU, an NTRU-base cryptosystem, has been proposed. The underlying algebraic structure for QTRU is the quaternion algebra which is a non-commutative algebra. Given the serious challenges ahead of quaternionic lattices, as well as the difficulties imposed by non-commutative algebra in solving (linear or non-linear, single-variable or multi-variable) equations, the proposed system proves more resistant against lattice attacks when compared to NTRU. Use of a non-commutative algebraic structure, that has been proven to be highly resistant against lattice attacks, accounts for the main strength of QTRU.

The general and necessary calculations regarding probability of successful decryption, message and key security, and message expansion have been presented for QTRU and the results have been compared to NTRU. Additionally, a group of generic parameters for QTRU have been introduced. Although the proposed method seems to be some four times slower than NTRU in totally equal conditions (i.e., selection of the same parameters for both NTRU and QTRU cryptosystems), QTRU is more resistant to lattice-based attacks when compared to NTRU. Hence, one can catch up on the speed by reducing the dimensions and still obtain the same level of security.

The following are other positive characteristics of QTRU:

- QTRU is totally compatible with NTRU and if coefficients of i, j, and k, are chosen to be equal to zero in all calculations, QTRU simply converts to NTRU. Therefore, QTRU is downward compatible to NTRU without any expense. This compatibility (just like the compatibility between 3DES and DES) can account as a highly positive point for QTRU.

- The data are encrypted four by four (four blocks by four blocks). As a result, an encrypted message may include four messages from a single source or four independent messages from four different sources. This characteristic may be very useful in protocol design or such applications as electronic voting, financial transactions and the like.

9 Further work

Over and above the discussion on cryptography, quaternionic lattice theory has valuable usages in coding theory, space-time coding in particular, as well as quantum physics. Therefore, studying the nature of quaternionic lattices is of interest in continuation of this line of research. Furthermore, NTRU and QTRU are based on a common concept that does not depend on a certain underlying algebraic structure. Hence, this concept can be used on different types of rings, modules, and vector spaces, or different kinds of algebras (in the sense of Cayley-Dickson) in order to produce new NTRU-like cryptosystems and explore their possible advantages.

10 Acknowledgement

Authors of this article wish to extend their cordial thanks to Professor Hussein Haji-Abulhassan, Faculty of Mathematics at Shahid Beheshti University for his generous guidance in preparing this paper.

References

[1] J. Hoffstein, J. Pipher, and J. H. Silverman, *NTRU: A Ring-Based public-key Cryptosystem*, in Proceedings of the Third International Symposium on Algorithmic Number Theory: Springer-Verlag, (1998).

[2] D. V. Bailey, D. Coffin, A. Elbirt, J. H. Silverman, and A. D. Woodbury, *NTRU in Constrained Devices*, in Proceedings of the Third International Workshop on Cryptographic Hardware and Embedded Systems: Springer-Verlag, (2001).

[3] A. May and J. H. Silverman, *Dimension Reduction Methods for Convolution Modular Lattices*, in Revised Papers from the International Conference on Cryptography and Lattices: Springer-Verlag, (2001).

[4] N. Howgrave-graham, J. Hoffstein, J. Pipher, and W. Whyte, *On estimating the lattice security of NTRU*, NTRU Cryptosystems Technical Report, (2007).

[5] J. Pipher, *Lectures on NTRU encryption algorithm and digital signature scheme*, NTRU Cryptosystems Inc., www.ntru.com, (2002).

[6] R. Kouzmenko, *Generalizations of NTRU Cryptosystem*, MsC Thesis, Polytechnique, Montreal, Canada., (2006).

[7] A. May, *Cryptanalysis of NTRU*, Unpublished preprint, (1999).

[8] D. Coppersmith and A. Shamir, *Lattice Attacks on NTRU*, Advances in Cryptology, EUROCRYPT '97, Lecture Notes in Computer Science, Springer-Verlag, vol. 1233, pp. 52-61, (1997).

[9] J. H. Silverman, *Dimension-Reduced Lattices, Zero-Forced Lattices, and NTRU public-key Cryptosystem*, NTRU Cryptosystems Technical Report, Report #12 & #13, (1999).

[10] T. Meskanen, *On NTRU Cryptosystem*, PhD Dissertation, Department of Mathematics, University of Turku, Finland, (2005).

[11] J. H. Conway and D. A. Smith, *On quaternion and Octonions: Their Geometry, Arithmetic, and Symmetry*, A K Peters, Ltd., 2003.

[12] J. C. Baez, *The Octonions*,Bulletin of the American Mathematical Society, pp. 45-205, (2006).

[13] Z. Y. Sham, *Quaternion Algebras and Quadratic Forms*, MsC Thesis in Pure Mathematics, Waterloo, Ontario, Canada, (2008).

[14] T. Y. Lam, *A First Course in Noncommutative Rings*, vol. 131: Springer-Verlag, 1991.

[15] A. Menezes, P. V. Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography:* CRC Press, 1996.

[16] F. Zhang, *quaternion and matrices of quaternion* , Linear Algebra and its Applications, vol. 251, pp. 21-57, (1997).

[17] H. Aslaksen, *Quaternionic determinants*, The Mathematical Intelligencer, vol. 18, pp. 57-65, (1996).

[18] N. Howgrave-graham, J. H. Silverman, and W. Whyte, *A Meet-In-The-Middle Attack on an NTRU Private Key*, NTRU Cryptosystems Technical Report, (2002).

[19] J. M. Peirce, *Determinants of quaternion* , Bulletin American Mathematical Society, vol. 5, pp. 335-337, (1899).

[20] W. A. Rutledge, *quaternion and Hadamard Matrices*, Proceeding of the American Mathematical Society, vol. 3, no. 4, pp. 625-630, (1952).

[21] D. Janovska and G. Opfer, *Linear equations in quaternion* , Numerical Mathematics and Advanced Applications, Proceedings of ENUMATH, Springer, Berlin, Heidelberg, , (2005).

[22] D. Micciancio and S. Goldwasser, *Complexity of Lattice Problems: a cryptographic perspective*, Boston, Massachusetts: Kluwer Academic Publishers, 2002.