CQTRU: A Commutative Quaternions Rings Based Public key Cryptosystem

Dr. Nadia M.G. Alsaidi
Applied Sciences Department, University of Technology/Baghdad
Emails: nadiamg08@gmail.com
Dr. Ahmad T. Sadiq
Computer Science Department, University of Technology/Baghdad
Ali A. Majid
Applied Sciences Department, University of Technology/Baghdad

Received on:28/10/2015 & Accepted on:29/9/2016

ABSTRACT

In this paper, we propose a new version of the NTRU public key cryptosystem called CQTRU. It is a four-dimensional cryptosystem based on the commutative quaternion ring. This public key system has an ability to encrypt four sets of data in each session. Therefore, it gains the positive points and the strength of NTRU cryptosystems. The definition of the ring of CQTRU is introduced with the definition of its operations. The three phases of the new proposed system (key generation, encryption and decryption) are discussed in details, in addition to the decryption failure probability, key security and message security. Finally, the resistance of CQTRU's against lattice attack is investigated.

Keywords: public key Cryptography; NTRU; Lattice attack; Quaternion algebra; Commutative cryptosystem

INTRODUCTION

In 1998, Hoffstein et al. [1] introduced a new public key cryptosystem called NTRU. Its operations are taking place in the polynomial rings with coefficients in Z. The security of NTRU is based on conjecture hard lattice problem to find a shortest vector in a lattice. Therefore, the NTRU is considered to be lattice based cryptosystem.

NTRU takes $O(N^2)$ operations where it encrypts or decrypts a message with length N and this speed is considered to be faster than other well-known systems as RSA which takes O operations for the same length of message in NTRU, a drawback sometimes appears in NTRU that the decryption function fails to give the original message, so the system has a decryption failure, but this problem can be solved by choosing parameters in such a way that the probability of having a decryption failure be as small as possible [2].

Many researches have been published to improve NTRU, some of these researches focused on improving the security by replacing the based ring such as CTRU [3] that replaced the polynomial ring into the ring of complex numbers, GNTRU [4]the cryptosystem presented by Kouzmenko based on the Gaussian integers, MaTRU [5] the cryptosystem based on matrix polynomial rings, and OTRU [6] a multi–dimensional public key cryptosystem based on the non- associative rings. In 2015, Alsaidi et al. introduced CQTRU cryptosystem based on commutative quaternions algebra [7]. In 2015, Atani et al. introduced EEH, a GGH like public key system based on Eisenstein integers Z[_3], where Z[_3] is a primitive cube root of unity. They demonstrated an improvement of their proposed system has an improvement over GGH in terms of security and efficiency [8]. In 2016, Thakur and Tripathi introduced BTRU, a new like NTRU cryptosystem, which replaces Z by a ring of polynomial with one variable _ over a rational field. They conveyed faster than NTRU [9].

https://doi.org/10.30684/etj.34.6B.19
2412-0758/University of Technology-Iraq, Baghdad, Iraq

The rest of this paper is structured as follows: In section 2 a present a mathematical background for the Commutative Quaternions. The NTRU crypto system is presented in section 3. Section 4 describes the proposed scheme in detail and in

Section 5 we analyze the performance and security of the proposed scheme. Finally, some conclusions are listed in section 6.

The Commutative Quaternions

In a four dimensions vector space, a set of commutative quaternions is denoted by $CQ,$ and defined as :

is the set of real numbers and satisfy the following multiplication rules:

A commutative quaternion element can be represented in a matrix form by an isomorphic function S from CQ to N, where N is the space of matrices and as follow:-

$$\begin{bmatrix} \\ y & & & \\ \end{bmatrix} \quad \ldots(1)$$

is called the fundamental matrix of commutative quaternion $CQ.$

Operations

Assume that $q_0, q_1 \in CQ,$ such that

$_1k,$ and w is a scalar in arbitrary field, then the addition, multiplication and scalar multiplication of two commutative quaternions elements are defined as follows:

- **Addition**

.. (2)

- **Multiplication**

The multiplication in commutative quaternion[7] is denoted by $\bullet$ and defined as

$-x$ $-x$ $-z$

…(3)

- **Scalar multiplication**

For any scalar

$= w[t_0 + x_0 i + y_0 j + z_0 k]$ ..(4)

Theorem 1[8].

For each can be represented by complex matrix, and and q can be uniquely representative as where and

C, and C the set of complex numbers. Then $q = c_1 + jc_2$ implies that $\varphi(a) = \begin{pmatrix} c_1 & c_2 \\ c_2 & c_1 \end{pmatrix}$, φ is a bijective map.

Depending on the previous theorem we design a formula of finding the multiplicative inverse and as follow:

For any element $a = a_0 + a_1 i + a_2 j + a_3 k \in CQ$, the multiplicative inverse for a is $a^{-1} = b = b_0 + b_1 i + b_2 j + b_3 k = [b_0 = s(a_1 t - a_2 x), x_1 = s(a_1 x + a_2 t), y_1 = s(a_2 z - a_1 y)$, and $z_1 = s(a_1 z + a_2 y)]$

The NTRU cryptosystem

The *NTRU* public key cryptosystem was presented by Pipher, Hoffstein and Silverman in 1998 [1]. The system is basically based on the ring of the truncated polynomials of degree $N - 1$. *NTRU* depends on three parameters N, p and q, such that, N is positive integer, p and q are also positive integer not necessary prime but they must be relatively prime. The ring of truncated polynomials are denoted by $R_x = Z[x]/(x^N - 1)$, where the multiplication in the ring R_x is denoted by " $*$ " .The parameters (N, p, q) are used in *NTRU* as follows:

N represents the degree of the truncated polynomials in *NTRU.*

p, q are two integers used to reduce the coefficients of truncated polynomials, where p is the prime modulus used to generate the public key and to set the coefficients of the message in the interval ($\frac{-p}{2}, \frac{p}{2}$], also in the decryption process, and q is the large modulus used to reduce the coefficients of the truncated polynomials in the interval $[\frac{-q}{2}, \frac{q}{2}]$.

Key Generation Phase

The *NTRU* keys are generated by choosing two polynomials $f \in L_f$ and $g \in L_g$. The inverses of polynomial f, $f_p, f_q \in R$ are calculated, such that,

$$f_p * f \equiv 1\ (mod\ \ p) \quad \text{and}\ f_q * f \equiv 1\ (mod\ \ q) \qquad \ldots(5)$$

Note that, if f is not invertible in both $(mod\ p)$ and $(mod\ q)$, then a new polynomial f should be chosen.

Encryption Phase

Suppose there is a message $m \in L_m$, where the coefficients of m are reduced $(mod\ p)$. In encryption function, a random polynomial $r \in L_r$ should be generated, then the cipher text is calculated by

$$e = r * h + m\ (mod\ \ q) \qquad ..(6)$$

Decryption Phase

From the cipher text *e* which is encrypted by *NTRU* is decrypted as follows:

1- Compute

$$a = f * e(mod\ q) \qquad ..(7)$$

where the coefficients are reduced in the interval $(\frac{-q}{2}, \frac{q}{2}]$.

2- The message is recovered by calculating

$$m_1 = f_p * a\ (mod\ q) \qquad ..(8)$$

Then, the coefficients of m_1 are reduced into the interval $\left(\frac{-p}{2}, \frac{p}{2}\right]$.

The NTRU cryptosystem on Commutative Quaternions Ring

After the definition of commutative quaternions, with its algebraic properties, we turn our attention to design *NTRU* type cryptosystem. In the previous work that improved *NTRU* each one had reduced the new based ring in the underlying algebra to the *Dedekind domains* and truncated polynomials used as coefficients to ensure that there exists an efficient algorithm for computing the inverse of an invertible polynomial in $R_x = Z[x]/(x^N - 1)$, since in the *Dedekind domains* every prime ideal is a maximal ideal, the extended Euclidean algorithm is used to find the inverse of scalar in R_x.This ring still keeps the same original properties that had before the reducing.

Commutative quaternion ring can be applied to any arbitrary field K (or ring), instead of real numbers *R*. In the original ring, $i^2 = k^2 = -1, j^2 = 1$ and $ij = k$, whereas, in this work, we can define i, j and k as $i^2 = a, j^2 = b, k^2 = ab$ and $ij = k$. By this definition a general commutative algebraic system is defined. Assume K is an arbitrary field, A can be defined as the commutative quaternion over K as follow:

$$A = \{a + bi + cj + dk \mid a, b, c, d \in K, i^2 = a, j^2 = b, ij = k\}.$$

Clearly, if $a = -1$ and $b = 1$, and K is the field of real numbers *R*, the original definition of commutative quaternion is obtained based on the choice of a and b and the nature of the field K. Consider there are two rings R_{xp} and R_{xq}, then for any two commutative quaternion rings A_p and A_q to be used in *CQTRU*, they are defined as:

$$A_p = \{f_0 + f_1 i + f_2 j + f_3 k | f_0, f_1, f_2, f_1 \in R_{xp}, i^2 = k^2 = -1, j^2 = 1, ij = k\} \text{ and}$$
$$A_q = \{f_0 + f_1 i + f_2 j + f_3 k | f_0, f_1, f_2, f_3 \in R_{xq}, i^2 = k^2 = -1, j^2 = 1, ij = k\}.$$

If $q_0, q_1 \in A_p (or\ A_q)$, where $q_0 = a_0 + b_0.i + c_0.j + d_0.k$ and $q_1 = a_1 + b_1.i + c_1.j + d_1k$.

Then, the operations on these two commutative quaternions (The addition, the multiplication and multiplicative inverse) are defined as follows:-

- **The addition**

$$q_0 + q_1 = (a_0 + a_1) + (b_0 + b_1)i + (c_0 + c_1)j + (d_0 + d_1) \quad \ldots(9)$$

- **The multiplication**

$$q_0 \centerdot q_1 = (a_0{*}a_1 - b_0{*}b_1 + c_0{*}c_1 - d_0{*}d_1) + (b_0{*}a_1 + a_0{*}b_1 + d_0{*}c_1 + c_0{*}d_1)i + (a_0{*}c_1 + c_0{*}a_1 - b_0{*}d_1 + d_0{*}b_1)j + (a_0{*}d_1 + d_0{*}a_1 + b_0{*}c_1 + c_0{*}b_1)k. \quad ..(10)$$

Where • and * denote the commutative quaternion multiplication and the convolution product respectively.

- **The multiplicative inverse**

The formula for finding the multiplicative inverse of commutative quaternion element is came from derivation based on theorem 1, let $q_0 = a_0 + a_1 i + a_2 j + a_3 k \in A$ then if q_0 has multiplicative inverse will be defined as follow:

let $[{a_0}^2 - {a_1}^2 - {a_2}^2 + {a_3}^2] = \alpha,$ $[2a_0a_1 - 2a_2a_3] = \beta$ and $\frac{1}{\alpha^2+\beta^2} = \delta$

Let ${q_0}^{-1} = q_1 = a_1 + a_1 i + a_1 j + a_1 k$ be the multiplicative inverse of q_0 then

$$[a_0 = \delta(a_0\alpha - a_1\beta), a_1 = \delta(a_1\alpha + a_0\beta), a_1 = s(a_2z - a_1y), \text{ and } z_1 = s(a_1z + a_2y)]$$

Key Generating Phase

For the public key, the user has to choose two small commutative quaternion (commutative quaternion with small α and β). F and G are two commutative quaternions that are generated randomly such that

$$F = f_0 + f_1 i + f_2 j + f_3 k \in L_F.$$
$$G = g_0 + g_1 i + g_2 j + g_3 k \in L_G.$$

The commutative quaternion F must be invertible over A_p and A_q. Whereas F is invertible over $A_p$and A_q if the polynomial $(\alpha^2 + \beta^2)$ is invertible in both rings R_{xp} and R_{xq}. Otherwise, a new commutative quaternion can be easily generated. The inverses of F are denoted by F_p and F_q respectively. Now, the public key is calculated as follows

$H = F_q \cdot G \mod q$

$= (f_{q0} * g_0 - f_{q1} * g_1 - f_{q2} * g_2 - f_{q3} * g_3) + (f_{q1} * g_0 + f_{q0} * g_0 + f_{q2} * g_3 + f_{q3} * g_2)i + (f_{q0} * g_2 + f_{q2} * g_0 - f_{q1} * g_3 - f_{q3} * g_1)j + (f_{q0} * g_3 + f_{q3} * g_0 + f_{q2} * g_1 + f_{q1} * g_2)k$.

…(11)

The commutative quaternions F, F_p and F_q are kept secret to be used in the decryption phase. It is obvious that, the estimated time to generate a key for the proposed scheme is 16 times slower than that of *NTRU*, when the same parameters (N, p, q) are chosen for both of them. However, to overcome this problem we can work with a lower dimension to achieve high execution time and the same security level. As it is mentioned before, the new system is a four dimension space. Hence, if one chooses the coefficients of i, j and k all are zeros in the commutative quaternions system then the system will be completely similar to *NTRU*. Also, if the coefficients of j and k are zero, a cryptosystem based on complex numbers is obtained. Finally, if the coefficients of i or j or k equal to zero, a tri-dimension scheme results.

Encryption Phase:

The encryption process is begun by generating of a random commutative quaternion called the blinding quaternion. The message should also be converted to commutative quaternion elements, where each of the four coefficients is a small polynomial and as follows:

$M = m_0 + m_1 i + m_2 j + m_3 k$ *where* m0, m_1, m_2, $m_3 \in L_M$.

Then, generate a random Commutative Quaternion $R = r_0 + r_1 i + r_2 j + r_3 k$, where $r_0, r_1, r_2, r_3 \in L_R$.

The encryption function is given by

$$E = p.H \cdot R + M \ (\text{mod } q) \qquad \ldots(12)$$

The Complexity of the Encryption Phase

In the encryption process, one commutative quaternion multiplication is needed which includes 16 convolution multiplication with $O(N^2)$ complexity, and 4 polynomial addition with $O(N)$ complexity. In encryption phase, the coefficients of the polynomials should be chosen in the interval$(\frac{-p}{2}, \frac{p}{2}]$, (in other words, m0, m1, m2 and m3 are small polynomials(mod q)). In this phase, at the same time, we encrypt four sets of data.

Decryption Phase

After receiving the commutative quaternion E, the following steps should be performed first, multiply E by the private key F such that,

$$B = F \cdot E \mod q \qquad \ldots (13)$$

Since B is computed$(mod\ q)$, after multiplying the coefficients of the four polynomials of B it should be reduced $(mod\ q)$ within the interval$(\frac{-q}{2}, \frac{q}{2}]$.

Now to get the original message$M$, the commutative quaternion D is calculated as follows:

$$D = F_p \cdot B \mod p. \qquad \ldots(14)$$

To have the original message, the commutative quaternion should be reduced into the interval $\left(\frac{-p}{2}, \frac{p}{2}\right]$.

Analysis

The proposed cryptosystem *CQTRU* is presented with all its mathematical theories. Now, in this section, the *CQTRU* features, implementation and analysis are discussed.

The *CQTRU* Implementation

Both *CQTRU* and *NTRU* are implemented in order to prove the efficiency and performance of the proposed system compared to original *NTRU* cryptosystem. The system with its three phases was implemented using Matlab on a PC with 2.4 GHZ Intel Core 3, Quad processor and 4 MB Ram under Windows 7 operation system.

1. A key generating phase and the encryption phase are implemented through the same user interface shown in Figure (5.1)
2.

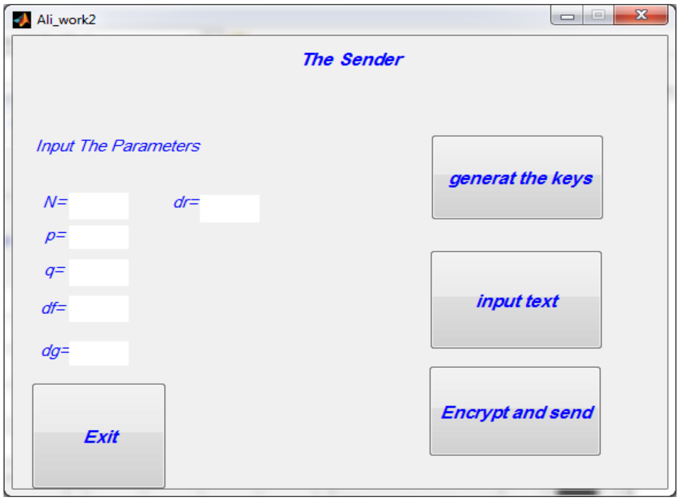

Figure 1 User interface for the sender

The input parameters to this widow are the public parameters N, p, q, d_f, d_g and d_r in addition to the message that the user wants to encrypt.

The public and private key are generated through one button, where the encryption cipher text with the public parameters N, p, q and the public key H to construct the message are shown in Figure 2

3. The decryption process is performed through another widow. In this widow.

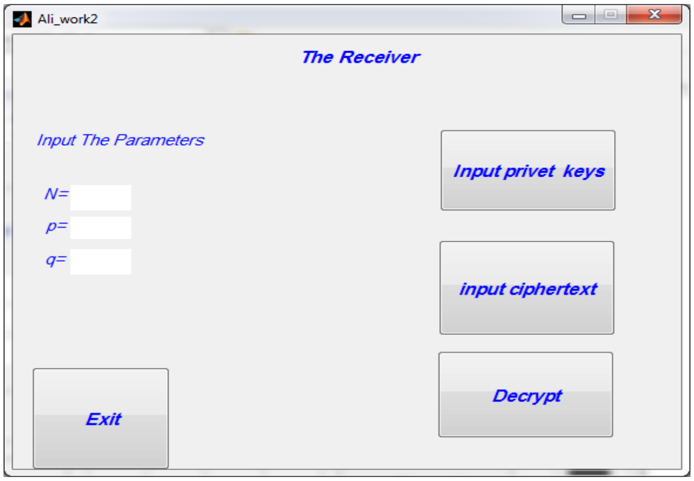

Figure 2 User interface for the receiver

The forms of *M, F, G, R*

In providing *CQTRU* four subsets of the commutative quaternions algebra A are used, these subsets are; $L_M, L_F, L_G,$ and L_R, each subset contains polynomials which have the following properties

- L_M is the subset that represents the space of polynomials of the message M. It contains all polynomials with coefficients between $\frac{-p}{2}$ and $\frac{p}{2}$.
- L_F is the subset that represents the space of the polynomials used for the private key F. Each polynomial in L_F has coefficients $\{-1, 0, 1\}$ determined by the public parameter d_F, where d_F is denotes the number of $+1's$ in each polynomial $f_i (i = 0, 1, 2, 3)$ and in each f_i there are (d_F) of $+1's$, $(d_F - 1)$ of $-1's$ coefficients and the remaining coefficients are $0's$.
- L_G and L_R are two subsets which represent the spaces of polynomials used for selecting the random commutative quaternions $G = g_0 + g_1 i + g_2 j + g_3 k$, where $g_0, g_1, g_2, g_3 \in L_G$ and $R = r_0 + r_1 i + r_2 j + r_3 k$ where $r_0, r_1, r_2, r_3 \in L_R$. Each polynomial in G has d_G coefficients belongs to $\{1, -1\}$ and the remaining are $0's$. Also, each polynomial in R has $d_R$coefficients belongs to $\{1, -1\}$ and the remaining are $0's$. Note that $d_F, d_G,$ and d_R are public parameters.

Decryption Failure

When the receiver decrypts a message that has been encrypted by *CQTRU*, he/she will be sure of getting of the original message, if all commutative quaternion coefficients of $F \bullet E = (pG \bullet R + F \bullet M)$ are within the interval $(\frac{-q}{2}, \frac{q}{2}]$, otherwise it will cause a decryption failure. Calculating the probability of successful decryption in *CQTRU* follow the same way as that used in *NTRU* under the same assumption considered in [4]. It is known that from the first step in decryption phase the commutative quaternion$A = pG \bullet F \bullet R + F \bullet M = a_0 + a_1 i + a_2 j + a_3 k$.

The coefficients of f_i, g_i and r_i are independent random variables take one of the values $\{-1, 0, +1\}$ randomly. Similar to *NTRU* the $var[a_{i,j}]$ is calculated. It is sufficient to assume that the mean in normal distribution probability is zero such that $E[f_{i,k,}] \approx 0$, $E[g_{i,k}]=E[r_{i,k}]=E[m_{i,k}]=0$ $E[a_{i,k}]=0$ where $i=0, 1, 2, 3$ and $k=0,\dots,N-1$.

A successful decryption means that all $a_{i,k}$*'s* lie within $(\frac{-q}{2}, \frac{q}{2}]$. Based on the assumption that $a_{i,k}$*'s* have normal distribution with zero mean and variance calculated in the same way as in *NTRU*, we have

$$Var[r_{i,k}.\, g_{j,l}] = \frac{4d_r.d_g}{N^2} \qquad \dots(15)$$

$$Var[f_{i,k}.\, m_{j,l}] = \frac{d_f(p-1)(p+1)}{6N} \qquad \dots(16)$$

$$Var[a_{0,k}] = \frac{16p^2 d_r d_g}{N} + \frac{4d_f(p-1)(p+1)}{6} \qquad \dots(17)$$

$$\Pr\left(|a_{i,k}| > \frac{q}{2}\right) = 2\ \Phi\left(\frac{q-1}{2\sigma}\right) - 1 \qquad \dots(18)$$

Where $\emptyset$ denotes the distribution of the standard normal variable and $\sigma=\sqrt{\frac{16\,p^2 drdg}{N} + \frac{4d_f\,(p-1)(p+1)}{6}}$. The $a_{i,k}$*'s* are assumed to be independent random variables. The decryption failure probability in *CQRTU* can be calculated by the following two observations

$$1 - \left(2\ \Phi\left(\frac{q-1}{2\sigma}\right) - 1\right)^N \qquad \dots\ (19)$$

$$1-\left(2\,\Phi\left(\frac{q-1}{2\sigma}\right)-1\right)^{4N} \qquad \ldots\ (20)$$

System Speed

It is obvious that, in *CQTRU* the variance of the coefficients ($pG \bullet R + F \bullet M$) increased by a factor of 4 than *NTRU* at the same parameters; this means that the probability for having a decryption failure is increased. But this probability is affected by the choice of the system parameters $d_r, d_g\ , d_f, q, p$ and N. This problem can be avoided or at least decreased according to the choice of the parametersd_f, d_g, and d_r to be large, but kept less than $\frac{N}{3}$, whereas, the speed of the system is not affected. In *CQTRU* public key cryptosystem the encryption has sixteen convolution product and thirty two in decryption. This seems to be slower than the *NTRU*, but in *CQTRU*, we can work in lower dimension and still keep a respected security level compared with *NTRU*.

Table 1 The encryption and decryption time in CQTRU for different message size

Message size in bytes	Encryption time in (S)	Decryption time in (S)
5	0.620	1.154
10	1.356	2.490
14	2.078	3.690
21	3.230	5.95
30	4.234	7.622
40	6.313	10.265

Security Analysis

In the security analysis of *CQTRU*, we will show that in terms of security, the resistance of *CQTRU* against the same attacks over *NTRU* discussed in [4] is four times better than that against these attacks in *CQTRU.*

1. Brute Force Attacks

In this attack the attacker tries all possible $\grave{f} \in L_f$ in an attempt to recover the private key f. He checks whether $f * h\ (mod\ q)$ has small entries. Another way is by trying all possible $\grave{g} \in L_g$ and checking if $g * h^{-1}\ (mod\ q)$ has small entries. In *CQTRU* the attacker uses the same procedure where the attacker knows the public parameters namely $d_r, d_g, d_f, q, p,$ and N, to look in the spaces L_F and L_G by using the same equations in [4] the attacker needs to seek in a space of large order as follows:

$$|\ L_F\ | \ = \binom{N}{d_f}^N \binom{N-\ d_f+1}{d_f} = \frac{(N!)^4}{\left(d_f!\right)^8\ \left(N-2\,d_f\right)!^4} \qquad \ldots(21)$$

$$|\ L_G\ | \ = \binom{N}{d_g}^N \binom{N-\ d_g+1}{d_g} = \frac{(N!)^4}{\left(d_g\,!\right)^8\ \left(N-2\,d_g\right)!^4} \qquad \ldots(22)$$

The space of L_F is a bigger that of L_G and for this reason, it is easier for the attacker to try search in L_G. An attacker can use the concept of brute force attack to get a message encrypted by *CQTRU* by searching in the space L_R, because, we know that $E = H \bullet R + M\ \ (mod\ q\ )$. If the attacker has an ability to find the blinding quaternion R, then he/she will be able to find the original message by calculating $M = E - H \bullet R\ (mod\ q)$. It is obvious that in brute force attack, the security of any message depends on the difficulty of finding R. The order of the space L_R is calculated in the same way that has been used to calculate the order of L_F and L_G and as follows:

$$|\, L_R \,| = \binom{N}{d_r}^N \binom{N - d_r + 1}{d_r} = \frac{(N!)^4}{(d_r!)^8 \;(N-2\, d_r)\,!^4} \qquad \ldots(23)$$

A comparison between the key security, message security in both systems *CQTRU* and *NTRU* shows that, the security of *CQTRU* is better than that of *NTRU.* In Tables 2 and 3, a comparison between the *CQTRU* and *NTRU* in terms of security for the key and the message is presented.

Table 2 Message space in CQTRU and NTRU

		Message Space	
N	d_R	*CQTRU*	*NTRU*
107	5	$6.1472 * 10^{63}$	$8.8546 * 10^{15}$
107	10	$3.8134 * 10^{106}$	$4.4190 * 10^{26}$
149	10	$1.1432 * 10^{119}$	$5.8147 * 10^{29}$
149	20	$3.9456 * 10^{190}$	$4.4568 * 10^{47}$
167	18	$3.5153 * 10^{186}$	$4.3300 * 10^{46}$
167	22	$9.2180 * 10^{211}$	$9.7985 * 10^{52}$
211	18	$5.4756 * 10^{202}$	$4.8374 * 10^{50}$
211	22	$1.9044 * 10^{232}$	$1.1747 * 10^{58}$
257	18	$1.2769 * 10^{216}$	$1.0630 * 10^{54}$
257	24	$1.6641 * 10^{269}$	$1.1358 * 10^{66}$

From Table 2, the security of the message in *CQTRU* depends on the value of N and the public parameter d_r which determines the message space and it is obvious that at the same parameters *CQTRU* is more secure than *NTRU* when attacker use brute force attack against both systems.

Table 3 Key space in CQTRU and NTRU

		Key Space	
N	d_g	CQTRU	NTRU
107	12	$3.3696 * 10^{120}$	$1.3549 * 10^{30}$
107	20	$1.0421 * 10^{163}$	$5.6817 * 10^{40}$
149	12	$6.0452 * 10^{135}$	$8.8176 * 10^{33}$
149	25	$8.2800 * 10^{216}$	$1.6963 * 10^{54}$
167	18	$3.5153 * 10^{186}$	$4.300 * 10^{46}$
167	27	$1.0436 * 10^{239}$	$5.6837 * 10^{59}$
211	20	$8.5378 * 10^{217}$	$3.0397 * 10^{54}$
211	34	$7.0057 * 10^{245}$	$2.8931 * 10^{61}$
257	20	$8.5849 * 10^{232}$	$1.7117 * 10^{58}$
257	24	$1.6641 * 10^{264}$	$1.1358 * 10^{66}$

In Table 3, the key space of *CQTRU* is much bigger than key space in *NTRU*. From (22) and the above table it is obvious that key space depends on the value of N and the public parameter d_G in *CQTRU*.

1. Lattice Attack

In this section, lattice attack that is presented in [4] is applied to *CQTRU*. It is known that every commutative quaternion is isomorphism to a matrix called the fundamental matrix given in (1). The parameters of the system (d_f, d_g, d_r, p, q, N) are known to the attacker as well as the public key $H = F_q \bullet G = h_0 + h_1 i + h_2 j + h_3 k$. When the attacker try to find either F or G, the *CQTRU* cryptosystem is broken. Note that, h_0, h_1, h_2 and h_3 are polynomials of order N over Z. These polynomials can be represented as vectors over Z^N as follows:

$H= h_0+h_1i+h_2j+h_3k \cong [h_0\ h_1\ h_2\ h_3]$

$h_0=[h_{0,0}\ \ h_{0,1}\ \ldots\ldots\ldots\ldots\ldots\ldots h_{0,N-1}]$

$h_1=[h_{1,0}\ \ h_{1,1}\ \ldots\ldots\ldots\ldots\ldots\ldots h_{1,N-1}]$

$h_2=[h_{2,0}\ \ h_{2,1}\ \ldots\ldots\ldots\ldots\ldots\ldots h_{2,N-1}]$

$h_3=[h_{3,0}\ \ h_{3,1}\ \ldots\ldots\ldots\ldots\ldots\ldots h_{3,N-1}]$.

For lattice analysis the polynomials h_0, h_1, h_2 and h_3 could be represented in their isomorphic representation, such that;

$$h(i)_{N\times N} = \begin{pmatrix} h_{i,0} & \cdots & h_{i,N-1} \\ h_{i,N-1} & & h_{i,N-2} \\ \vdots & \ddots & \vdots \\ h_{i,2} & \cdots & h_{i,1} \\ h_{i,1} & & h_{i,0} \end{pmatrix} \quad i=0,\ 1,\ 2,\ 3 \qquad \ldots (24)$$

Where $h(i)_{N\times N}$ is called the circulant matrix [8]. From the assumptions above, the description of the *partial lattice attack* is given by: Let us denote the commutative quaternions F and G as $F=[f_0\ \ f_1\ \ f_2\ \ f_3]$ and $G=[g_0\ \ g_1\ \ g_2\ \ g_3]$ where $f_0, f_1, f_2, f_3, g_0, g_1, g_2, g_3 \in Z[x]/(x^N - 1)$, this lattice denoted by $L_{partial}$ is defined as follows:

$$L_{partial} = \begin{pmatrix} I_{4N\times 4N} & 0_{4N\times 4N} \\ H_{4N\times 4N} & qI_{4N\times 4N} \end{pmatrix} \in Z^{8N}$$

Where I denotes the identity matrix, 0 denotes the zero matrix, and H is the fundamental matrix for the h_i's satisfied$F \bullet H = G$. Since the points generated by *CQTRU* lattices include a partial subset of the total set of vectors that satisfy $F \bullet H=G$, this leads to a major difference between NTRU and CQTRU lattices.

The attacker will use the lattice reduction algorithm to find a short vector which satisfies $F \bullet H = G$. However, even with such a promising assumption the dimension of $L_{partial}$ is $8N \times 8N$ while the dimension of the lattice in *NTRU is* $2N \times 2N$, this means that the *LLL reduce* algorithm applied to $L_{partial}$ takes time larger than the one that applied in the lattice of *NTRU* with four times factor, and the short vectors in $L_{partial}$ will not always guarantee being found. When *CQTRU* deals with parameters (*N=107, p, q*), it gives the same result as *NTRU* with (*N=428, p, q*). For any chosen (*N, p, q*), when these parameters used in *CQTRU* the system will be four times slower than *NTRU*, but *CQTRU* has a security equal to the security of *NTRU* with (*4N, p, q*) dimensions. However, the security of *CQTRU* is four times larger than the security. This leads to deduce that *CQTRU* with small dimension have security advantage over *NTRU.*

In pursuit, the lattice attacks do not always give a successful results, because generally $L_{partial}$ does not necessarily contains all answers of $F \bullet H=G$, in such a way that $f_0, f_1, f_2, f_3, g_0, g_1, g_2, g$ or g_3 would be short vectors. Therefore, the attacker must find a lattice contain all vectors satisfying the congruence of $F \bullet H=G$.

CONCLUSIONS

This paper focuses on the *NTRU* public key cryptosystem features, it is proposed new *NTRU*-like cryptosystem called *CQTRU* based on four dimensional space ring called the commutative quaternions algebra. All mathematical rules, features of the new system and the comparisons between *CQTRU* and *NTRU* have been presented. These comparisons and discussions on the new system are summarized. The *CQTRU* basic algorithm has succeeded in all the trials; there is no percentage of failure at all security levels. In *CQTRU* the inverse algorithm gives the feature for taking a private key of two or three dimensions keeping the system working in four dimensions for the calculations of the public key, so it gives a simple way for the choose of private key and keep the security high. Unlike *NTRU,* in *CQTRU* one can take small dimensions for polynomial and still have a respective level of security and from this feature one can conclude that, if the user wishes to work with faster system and keep the security at good level, *CQTRU* is batter to use than *NTRU*, *CQTRU* is more resistant to alternate key attack and brute force attack than *NTRU* because of its big spaces compared to those in *NTRU.* The resistance of *CQTRU* to lattice attack is at least four times better than that of *NTRU* at the same dimension and also in *CQTRU* even with the use of *LLL* reduce algorithm an attacker cannot always guarantee to find the private key. In *CQTRU,* we can encrypt and decrypt four messages at the same time, because it has four dimensions and this property provides efficient computation time.

REFERENCES

[1]J. Hoffstein, J. Pipher, and J. H. Silverman, “NTRU: A ring-based public key cryptosystem,” Algorithmic Number Theory in Lecture Notes in Computer Science. Springer Verlag, 1998, pp. 267–288.

[2] J. Hoffstein, J. Pipher and J. H. Silverman, “An Introduction to Mathematical Cryptography”, Undergraduate Texts in Mathematics, Springer, 2008.

[3]P. Gaborit, J. Ohler,, Sole, P, “CTRU, a Polynomial Analogue of NTRU”, INRIA. Rapport de recherche, N.4621 November 2002.

[4]R. Kouzmenko,"Generalizations of the NTRU Cryptosystem", Master’s thesis, Polytechnique, Montreal, Canada, 2006. `

[5]M. Coglianese, B. Goi , "MaTRU: a new NTRU-based cryptosystem". Indocrypt 2005. Lecture Notes in Computer Science, vol. 3797, pp. 232–243.

[6]E. Malekian and A. Zakerolhosseini, "OTRU: A Non-Associative and High Speed Public Key Cryptosystem", 15th Computer Society of Iran (CSI) Symposium on Computer Architecture and Digital Systems, CADS 2010, pp.83-90.

[7]F. Catoni, R. Cannata and P. Zampetti, "An Introduction to Commutative Quaternions", Adv. appl. Clifford alg. vol.16, 2006, pp. 1–28.

[8]E. Malekian, A. Zakerolhosseini and A. Mashatan," QTRU: A Lattice Attack Resistant Version of NTRU PKCS Based on Quaternion Algebra", IACR Cryptology, 2009.

[9]N.Alsaidi.M.Said.,A.T.Sadiq,and A.A.Majeed,An improved NTRU Cryptosystem via Commutative Quaternions Algebra,Int’l Conf.Security and Management SAM’15,2015,PP.198-203.

[10]A. R. Ebrahimi., S. Ebrahimi, and K. Amir Hassani, EEH: AGGH-Like Public Key Cryptosystem Over The Eisenstein Integers Using Polynomial Representations, ISeCure; July 2015, Vol. 7 Issue 2, p1.

[11]K. Thakur and B. P. Tripathi, BTRU, A Rational Polynomial Analogue of NTRU Cryptosystem, International Journal of Computer Applications, Foundation of Computer Science (FCS), NY, USA, Vol. 145 – No.12. 2016.

[12]H. R. Yassein, and N. Alsaidi, HXDTRU Crytosystem Based On Hexadecnion Algebra, 5th International Cryptology and Information Security Conference, 2016. CRYPTOLOGY2016.