# PairTRU: Pairwise Non-commutative Extension of The NTRU Public key Cryptosystem

Amir Hassani Karbasi[1], Shahabaddin Ebrahimi Atani[2], and Reza Ebrahimi Atani[3]

[1] Department of Mathematics, University Campus 2, University of Guilan, Rasht, Iran. karbasi@phd.guilan.ac.ir
[2] Department of Mathematics, University of Guilan, Rasht, Iran. ebrahimi@guilan.ac.ir
[3] Department of Computer Engineering, University of Guilan, P. O. Box 3756, Rasht, Iran; rebrahimi@guilan.ac.ir. e-mail: amirhassanikarbasi@gmail.com

**Abstract**—We show a novel lattice-based scheme (PairTRU) which is a non-commutative variant of the NTRU. The original NTRU is defined via the ring of quotient with variable in integers and this system works in the ring $R = \frac{\mathbb{Z}[x]}{<x^N-1>}$. We extend this system over $\mathbb{Z} \times \mathbb{Z}$ and it performs all of operations in the non-commutative ring $\mathbf{M} = \frac{M(k,\mathbb{Z}\times\mathbb{Z})[x]}{<(I_{k\times k},I_{k\times k})x^N-(I_{k\times k},I_{k\times k})>}$, where $\mathbf{M}$ is a matrix ring of $k \times k$ matrices of polynomials in $R = \frac{(\mathbb{Z}\times\mathbb{Z})[x]}{<(1,1)x^N-(1,1)>}$. In PairTRU, encrypting and decrypting are non-commutative and the cryptosystem is secure for linear algebra and Lattice-based attacks. PairTRU is designed using the NTRU core and reflects high levels of security by two-sided matrix multiplication with pairwise entries.

**Keywords**—Public key cryptography, Lattice-based cryptosystem, NTRU, Matrix rings.

## 1. Introduction

Lattice-based cryptographic structures are known with their worst-case hardness for strong security proofs and relatively efficient implementations for post-quantum cryptography which have considerable active research area. Moreover, lattice cryptography is compatible and secure for quantum computers [1]. Our focus here will be mainly on the theoretical aspects of lattice-based cryptography.

J. Hoffstein, J. Pipher and J. Silverman presented NTRU public key scheme. The first version of the NTRU cryptosystem was presented at the Crypto ‘96 conference [2]. The mathematical background of these lattice-based systems lies in polynomial algebra. The fundamental operation is basen on two modulos for reducing of polynomials. The NTRU encryption scheme (NTRUEncrypt), for gaining a high efficiency, utilizes the structured lattices and their properties and that makes it a potential practical system. It’s core is different from RSA and ECC, and it is more efficient than those schemes. It is considerably faster than RSA and ECC or any other public key system. Statistics [3] shows that the security level of NTRU with $N = 251$, RSA with 1024 bits, and ECC [4] with 163 bits are comparable. The experimental results can be seen in table 1. The security of the NTRU is based on intractability of solving “Shortest Vector Problem / Closest Vector Problem” (SVP/CVP) in a particular type of lattice called $Convolutional\ Modular\ Lattices$ (CML) related to the cyclotomic ring $R = \frac{\mathbb{Z}[x]}{<x^N-1>}$. Thus, lattice

TABLE 1
A comparison of NTRUEncrypt, RSA and the elliptic curves cryptosystem made using a 800MHz Pentium III computer [5].

| | NTRU 251 | RSA 1024 | ECC 163 |
|---|---|---|---|
| Public key(bits) | 2008 | 1024 | 164 |
| Private key(bits) | 251 | 1024 | 163 |
| Plaintext block(bits) | 160 | 702 | 163 |
| Ciphertext block(bits) | 2008 | 1024 | 163 |
| Encryption speed(blocks/sec) | 22727 | 1280 | 458 |
| Encryption speed(Mbits/sec) | 3.6 | 0.9 | 0.075 |
| Decryption speed(blocks/sec) | 10869 | 110 | 702 |
| Decryption speed(Mbits/sec) | 1.7 | 0.077 | 0.11 |

reductions are important attacks for NTRU [6] and Chinese Remainder Theorem (CRT) attacks [7]. Lattice attacks want to find a key for decrypting. In NTRU, without FFT, the multiplication of two polynomials has $O(N^2)$ cost. Moreover, we can use optimization processes for multiplications such as FFT or NTT. Also, the key generation of NTRU is very fast, as well as it utilizes vector multiplications (convolutions), so the NTRU has an appropriate speed. Standardization for the NTRU has now been with version IEEE P1363.1 [8]. The cryptanalysis results for NTRU can be found in [7], [9], [10], [11], [12], [34]. Improvements of the security of NTRU as well as some variants are described using polynomial rings with special type of coefficients such as $GF(2^k)[x]$ [13], the non-commutative matrix ring of polynomials in $\frac{\mathbb{Z}[x]}{<x^N-1>}$ [14], the non-commutative matrix ring $\mathbf{M} = \frac{M_k\mathbb{Z}[x]}{X^n - I_{k\times k}}$, where $\mathbf{M}$ [15], Dedekind domains such as $\mathbb{Z}[i]$, $\mathbb{Z}[\sqrt{-2}]$, $\mathbb{Z}[\zeta_3]$ and $\mathbb{Z}[\zeta_5]$ [16], [17], [18], QTRU, based on Quaternion algebra [19] and authors' lattice-based schemes [20], [27], [28], [29], [30], [31], [32], [33]. Here, we extend the core of NTRU to a broader algebra than $\mathbb{Z}$ such as $M(k, \mathbb{Z}\times\mathbb{Z})$ matrices. The NTRU can be summarized as follows:

Let $R = \frac{\mathbb{Z}[x]}{<x^N-1>}$ with $N$ prime. Assume $q$ be an integer and assume $p \in R$ co-prime with $q$ and such that the plaintext space $R/ < p >$ is large. In general, we can take $p = 3$ or $p = x + 2$. As a private key, $(f, g) \in R^2$ randomly chosen with coefficient of $\{-1, 0, 1\}$ proportionally. For improved decryption efficiency, invertible $f \equiv 1 \bmod p$ for modulo $q$ and modulo $p$ is chosen, therefore, the public-key is computed $h = pg * f^{-1} \bmod q$. For encrypting $m \in R/ < p >$, we can choose a ephemeral element $r \in R$ with short norm and gives the ciphertext $e \equiv hr + m \bmod q$. To decrypt the ciphertext $e$, one can compute $f * e \bmod q$. If $e$ be a honest data, this gives $pgr + fm \bmod q$. Since $p, g, r, f, m$ have small coefficients, we can show that using reducing technique modulo $q$, we have $pgr + fm \in R$. Next, by reducing modulo $p$, it provides $fm$ mod $p$. Now, we can multiply the previous result by the inverse of $f$ modulo $p$. In addition, encryption is probabilistic and decryption failure can be shown for some sets of parameters. Therefore, we can decrease or eliminate decryption failure with appropriate parameters.

Here, another version of NTRU is proposed, which is called PairTRU. Our proposed cryptosystem is a variant based on non-commutativity rule over $\mathbb{Z}_q$, i.e., over the quotient polynomial ring $\frac{\mathbb{Z}_q[x]}{<x^N-1>}$. The PairTRU cryptosystem performs an efficient multiplication while providing a security level higher than that of NTRU. It operates in the ring of $k \times k$ matrices with pairwise entries of $k^2$ different polynomials in $R = \frac{(\mathbb{Z}\times\mathbb{Z})[x]}{<(1,1)x^N-(1,1)>}$. Here, matrix operations and multiplications in PairTRU is based on non-commutativity rules. Attackers must recognize left and right computations. Also Shamir [6] showed that non-commutative operations give a high level of security against Lattice attacks. In addition, we can construct more complicated cryptosystems using non-commutative matrix rings. By replacing $nk^2 = N$, comparing of instances are easy. For encrypting and decrypting in NTRU, we have $O(N^2)$ or $O(n^2k^4)$ operations for element $N$ but in PairTRU for identical levels, we have $\tilde{O}(nlogn.k^{2.376})$ operations using Fast Fourier Transformation for linear transformation [15]. We use $\tilde{O}(.)$ to hide poly-logarithmic factors. Furthermore, we can inverse these kind of polynomials by some methods like [21].

The rest of this paper is structured as follows: In section 2, we give some notations and norm estimations. In section 3, we suggest an NTRU-Like public key cryptosystem (PairTRU) using non-commutative matrix rings with pairwise entries. Analysis of security for our proposed PairTRU cryptosystem is described in sections 4. Section 5 is dedicated to efficiency and comparisons. Finally, conclusions is given in section 6.

## 2. Notations and Definitions

Set $R = \frac{(\mathbb{Z}\times\mathbb{Z})[x]}{<(1,1)x^N-(1,1)>}$, an element $f$ in the ring $R$ can be represented as follows:

$$f = \{(a_0, b_0)+(a_1, b_1)x+...+(a_{N-1}, b_{N-1})x^{N-1} | a_i, b_i \in \mathbb{Z}\}.$$

Since the elements of $R$ are polynomials of degree less than $N$, they can be represented as vectors. We can use the following representation of $f \in R$ interchangeably when there is no possibility of confusion.

$$f = \sum_{i=0}^{N-1} (a_i, b_i)x^i = \{(a_0, b_0), (a_1, b_1), ..., (a_{N-1}, b_{N-1})\}$$
$$\in (\mathbb{Z}\times\mathbb{Z})^N.$$

By reducing an element of $R$ modulo $(p, p)$ or $(q, q)$, we mean reducing of its pairwise coefficients modulo $p$ or $q$ as follows (i.e., [$a_i$ mod $p$, $b_i$ mod $p$] or [$a_i$ mod $q$, $b_i$ mod $q$]) that we denoted by componentwise mod $(p, p)$ or mod $(q, q)$.

$$R_{(p,p)} = \frac{(\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/p\mathbb{Z})[x]}{< (1,1)x^N - (1,1) >},$$

$$R_{(q,q)} = \frac{(\mathbb{Z}/q\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z})[x]}{< (1,1)x^N - (1,1) >}.$$

We can compute summation of two elements $f$ and $g$ of $R$ as follows:

$$f = \{(a_0, b_0)+(a_1, b_1)x+...+(a_{N-1}, b_{N-1})x^{N-1}|a_i, b_i \in \mathbb{Z}\},$$

$$g = \{(c_0, d_0)+(c_1, d_1)x+...+(c_{N-1}, d_{N-1})x^{N-1}|c_i, d_i \in \mathbb{Z}\},$$

$$\begin{aligned} f + g =&\{(a_0 + c_0, b_0 + d_0) + (a_1 + c_1, b_1 + d_1)x + ... \\ &+ (a_{N-1} + c_{N-1}, b_{N-1} + d_{N-1})x^{N-1} \\ &|a_i, b_i, c_i, d_i \in \mathbb{Z}\}. \end{aligned}$$

Clearly, we have:

$$\begin{aligned} f - g =&\{(a_0 - c_0, b_0 - d_0) + (a_1 - c_1, b_1 - d_1)x + ... \\ &+ (a_{N-1} - c_{N-1}, b_{N-1} - d_{N-1})x^{N-1} \\ &|a_i, b_i, c_i, d_i \in \mathbb{Z}\}. \end{aligned}$$

Also we can compute convolution multiplication of two elements $f$ and $g$ of $R$, denoted by $f * g$, as follows. At first, we show the ordinary convolution multiplication for $k$-th coefficient, then we reflect pairwise convolution multiplication.

$$\begin{aligned} f * g &= \sum_{i=0}^{k} f_i.g_{k-i} + \sum_{i=k+1}^{N-1} f_i.g_{N+k-i} \\ &= \sum_{i+j \equiv k \ (mod\ N)} f_i.g_j \end{aligned}$$

Now we can change the usual convolution multiplication to coordinatewise multiplication for two entries in coefficients. Clearly, we compute pairwise convolution multiplication as follows:

$$f * g = \sum_{(i,i)+(j,j) \equiv (k,k) \ (mod\ (N,N))} (a_i, b_i).(c_{k-i}, d_{k-i}) \tag{2.1}$$

**Definition 2.1.** We would like to show that $(\mathbb{Z} \times \mathbb{Z}, \leq')$ is a partially ordered set (poset). So we define $(a_i, b_i) \leq' (a'_i, b'_i) \Leftrightarrow [a_i \leq a'_i \ and \ b_i \leq b'_i]$.

1 Reflexivity:
$\forall (a_i, b_i) \in (\mathbb{Z} \times \mathbb{Z})$ :
$(a_i, b_i) \leq' (a_i, b_i) \Leftrightarrow [a_i \leq a_i$ and $b_i \leq b_i]$.

2 Antisymmetry:
$\forall (a_i, b_i), (c_i, d_i) \in (\mathbb{Z} \times \mathbb{Z})$ :
$[(a_i, b_i) \leq' (c_i, d_i)$ and $(c_i, d_i) \leq' (a_i, b_i)] \Rightarrow [a_i \leq c_i$ and $b_i \leq d_i]$ and $[c_i \leq a_i$ and $d_i \leq b_i] \Rightarrow (a_i, b_i) = (c_i, d_i)$.

3 transitivity:
$\forall (a_i, b_i), (c_i, d_i), (e_i, f_i) \in (\mathbb{Z} \times \mathbb{Z})$ :
$[(a_i, b_i) \leq' (c_i, d_i)$ and $(c_i, d_i) \leq' (e_i, f_i)] \Rightarrow [a_i \leq c_i$ and $b_i \leq d_i]$ and $[c_i \leq e_i$ and $d_i \leq f_i] \Rightarrow [a_i \leq e_i$ and $b_i \leq f_i] \Rightarrow (a_i, b_i) \leq' (e_i, f_i)$.

Hence as a consequent $(R, \leq')$ is poset.

**Definition 2.2.** We define $(a', b')$ is a maximum element in $f$ if $(c_i, d_i) \leq' (a', b')$ for all $(c_i, d_i)$ in $f$ and we show by: $\max_{1 \leq i \leq N} \ \{(a_i, b_i)\}$

**Definition 2.3.** We also define $(a'', b'')$ is a minimum element in $f$ if $(a'', b'') \leq' (c_i, d_i)$ for all $(c_i, d_i)$ in $f$ and we show by: $\min_{1 \leq i \leq N} \ \{(a_i, b_i)\}$

**Definition 2.4.** The width of an element $f \in R$ is defined to be:

$$|f|_\infty = \max_{1 \leq i \leq N} \ \{(a_i, b_i)\} - \min_{1 \leq i \leq N} \ \{(a_i, b_i)\} \tag{2.2}$$

**Definition 2.5.** The centered $L_\perp$ norm on $R$ is defined by: $|f|_\perp = \sqrt{\sum_{i=0}^{N-1} (a_i, b_i)^2 - \frac{1}{N}(\sum_{i=0}^{N-1}(a_i, b_i))^2} \Rightarrow (a_i, b_i).(a_i, b_i) = (a_i a_i, b_i b_i) = (a_i^2, b_i^2) \Rightarrow \sum_{i=0}^{N-1}(a_i^2, b_i^2) = (a_0, b_0)^2 + (a_1, b_1)^2 + ... +(a_{N-1}, b_{N-1})^2 = (a_0^2, b_0^2) + (a_1^2, b_1^2) + ... + (a_{N-1}^2, b_{N-1}^2) = (a_0^2 + a_1^2 + ... + a_{N-1}^2, b_0^2 + b_1^2 + ... + b_{N-1}^2) \Rightarrow \frac{1}{N}(a_i, b_i) = (\frac{a_i}{N}, \frac{b_i}{N}) \Rightarrow (\sum_{i=0}^{N-1}(a_i, b_i))^2 = ((a_0, b_0) + (a_1, b_1) + ... +(a_{N-1}, b_{N-1}))^2 = (a_0 + a_1 + ... + a_{N-1}, b_0 + b_1 + ... + b_{N-1})^2 = [(a_0 + a_1 + ... + a_{N-1})^2, (b_0 + b_1 + ... + b_{N-1})^2] \Rightarrow \sum_{i=0}^{N-1}(a_i, b_i)^2 - \frac{1}{N}(\sum_{i=0}^{N-1}(a_i, b_i))^2 = [(a_0^2 + a_1^2 + ... + a_{N-1}^2, b_0^2 + b_1^2 + ... + b_{N-1}^2)] - [\frac{(a_0+a_1+...+a_{N-1})^2}{N}, \frac{(b_0+b_1+...+b_{N-1})^2}{N}] = (a_0^2 + a_1^2 + ... + a_{N-1}^2 - \frac{(a_0+a_1+...+a_{N-1})^2}{N}, \quad b_0^2 + b_1^2 + ... + b_{N-1}^2 - \frac{(b_0+b_1+...+b_{N-1})^2}{N})$

Now we set:

$$s = (a_0^2 + a_1^2 + ... + a_{N-1}^2 - \frac{(a_0 + a_1 + ... + a_{N-1})^2}{N},$$

and

$$t = b_0^2 + b_1^2 + ... + b_{N-1}^2 - \frac{(b_0 + b_1 + ... + b_{N-1})^2}{N}),$$

Therefore

$$(s,t) \Rightarrow \sqrt{(s,t)} = (s,t)^{\frac{1}{2}} = (s^{\frac{1}{2}}, t^{\frac{1}{2}}) = (\sqrt{s}, \sqrt{t}). \quad (2.3)$$

Note that if there is no maximum or minimum element in polynomial $f$, we can choose simply another polynomial $f$ of $R$. The norm is imposed componentwise on $(f,g) \in R^2 : |(f,g)|_\perp = |f|_\perp + |g|_\perp$. Note that $|f|_\perp$ is the standard deviation of the entries of $f$, times $\sqrt{N}$.

**Lemma 2.6.** *If the entries of $f$ have zero mean, the centered norm is the same as the Euclidean $L_2$ norm.*

*Proof:* Having zero mean implies that $\frac{1}{N}(\sum_{i=0}^{N-1}(a_i,b_i)) = 0$ so $|f|_\perp = \sqrt{\sum_{i=0}^{N-1}(a_i,b_i)^2 - 0} = ||f||_2$. □

**Definition 2.7.** Two norms $||.||_\alpha$ and $||.||_\beta$ on a vector space $R$ are called equivalent if there exist positive real numbers $\gamma_1, \gamma_2$ such that for all $f$ in $R$: $\gamma_1||f||_\alpha \leq ||f||_\beta \leq \gamma_2||f||_\alpha$. Hence for any $(0,0) \leq' \epsilon = (\lambda_1, \lambda_2)$ there are constants $(0,0) \leq' \gamma_1, \gamma_2$, rely on $\epsilon$ and $N$, such that for random $f, g \in R$, the probability is greater than $1 - \epsilon$ that they satisfy

$$\gamma_1|f|_\perp|g|_\perp \leq' |f * g|_\infty \leq' \gamma_2|f|_\perp|g|_\perp \quad (2.4)$$

We already observe that converting from $R$ to $R_(q,q)$ is clear and we can reduce the coefficients of a polynomial modulo $(q,q)$. This reduction is a ring homomorphism.

$$(f + g) \; mod \; (q,q) = (f \; mod \; (q,q)) + (g \; mod \; (q,q))$$

$$(f * g) \; mod \; (q,q) = (f \; mod \; (q,q)) * (g \; mod \; (q,q))$$

**Example 2.8.** Assume $N = 3$ and $(q,q) = (3,3)$. Let $f = (4,3) + (2,-1)x + (-5,-6)x^2 \in R$. We can simply reduce the pairwise coefficients of polynomial $f$ modulo $(q,q) = (3,3)$ as $f' = (1,0) + (2,2)x + (1,0)x^2 \in R_{(3,3)}$.

**Definition 2.9.** Let $f \in R_(q,q)$. The centered lift of $f$ to $R$ is the unique polynomial $f' \in R$ satisfying $f'$ mod $(q,q)$ = $f$ whose pairwise coefficients are chosen in the interval $-\frac{q}{2} < a'_i \leq \frac{q}{2}$ and $-\frac{q}{2} < b'_i \leq \frac{q}{2}$.

**Example 2.10.** Suppose $N = 3$ and $(q,q) = (7,7)$. Let $f = (5,3) + (3,-6)x + (2,4)x^2 \in R_{(7,7)}$. For the centered lift of the polynomial $f$ we sample the pairwise coefficients of polynomial $f'$ from $\{-3,-2,-1,0,1,2,3\}$ as $f' = (-2,3) + (3,1)x + (2,-3)x^2 \in R$.

**Proposition 2.11.** *A polynomial $f$ in $R$ is invertible mod $(q,q)$ if and only if gcd$(f, <(1,1)x^N - (1,1)>) \equiv (1,1)$ mod $(q,q)$.*

*Proof:* We know that $\frac{(\mathbb{Z}\times\mathbb{Z})[x]}{I} = \{t + I | t \in (\mathbb{Z}\times\mathbb{Z})[x]\}$. Set $f \in \frac{(\mathbb{Z}\times\mathbb{Z})[x]}{I}$, then $f = t + I$ for some $t \in (\mathbb{Z}\times\mathbb{Z})[x]$. We show that there exists $g = t' + I \in (\mathbb{Z}\times\mathbb{Z})[x]$ such that $f * g = (1,1) + I$.

$$\begin{aligned}
&t.t' + I = (1,1) + I \\
&\Rightarrow t.t' - (1,1) \in I \\
&\Rightarrow I = <(1,1)x^N - (1,1)> \\
&= \{h * ((1,1)x^N - (1,1)) | h \in (\mathbb{Z}\times\mathbb{Z})[x]\} \\
&\Rightarrow \exists u \in (\mathbb{Z}\times\mathbb{Z})[x] : \\
&\quad t.t' - (1,1) = u((1,1)x^N - (1,1)) \\
&\Rightarrow t.t' - u((1,1)x^N - (1,1)) = (1,1) \\
&\Rightarrow t.t' - u((1,1)x^N - (1,1)) \equiv (1,1) \; mod \; (q,q)
\end{aligned}$$

□

All arithmetics in PairTRU are given in $\mathbf{M} = \frac{M(k,\mathbb{Z}\times\mathbb{Z})[x]}{<(I_{k\times k}, I_{k\times k})x^N - (I_{k\times k}, I_{k\times k})>}$, such that $\mathbf{M}$ is $k \times k$ matrices with entries $R = \frac{(\mathbb{Z}\times\mathbb{Z})[x]}{<(1,1)x^N - (1,1)>}$.

**Definition 2.12.** Width of $f \in \mathbf{M}$ is defined by:

$$|f|_\infty = Max(Coef.inPolys.inf)$$
$$-Min(Coef.inPolys.inf)$$

We say a matrix $f \in \mathbf{M}$ is short if $|f|_\infty < (q,q)$. Similarly the polynomial $f \in R$ is said to be short if $|f|_\infty < (q,q)$.

## 3. PairTRU Cryptosystem

### 3.1. Parameter Creation

PairTRU scheme utilizes five positive integer $(n,k,p,q,d)$ where $p \ll q$ are co-prime and five sets of matrices $(L_f, L_c, L_r, L_w, L_m) \subset \mathbf{M}$. The set of matrices

$(L_f, L_c, L_r, L_w)$ are chosen from $L(d_1, d_2)$ as follows:

$$L(d_1, d_2) := \{f \in R | f\ has\ d_1\ coef.\ equal\ 1,$$

$$d_2\ coef.\ equal\ -1,\ and\ rest\ 0\}$$

where, $d = d_1 = d_2 \approx n/p$.

We express the space of message $L_m$ as follows:

$$L_m := \{m \in \mathbf{M} | Polys.\ in\ m\ has\ coef.$$

$$lying\ between\ -\frac{p-1}{2}\ and\ \frac{p-1}{2}\}$$

We now describe all five sets of matrices $(L_f, L_c, L_r, L_w, L_m) \subset \mathbf{M}$:

- invertible $L_f$ with entries $f$ and $g$ gives private-key and $L_r$ with ephemeral entry $r$ is chosen for encrypting.
- public key entries $c$ and $w$ are in matrix set $L_c$ as invertible modulo $(p, p)$ and $L_w$ respectively.
- We sample $m \in L_m$ as a plaintext.

### 3.2. Key Generation

Suppose that Bob wants to create his public and private key. He randomly chooses $f, g \in L_f$ and $w \in L_w$ and $c \in L_c$. We denote the inverses of $f, g$ and $c$ modulo $(p, p)$ and modulo $(q, q)$ by notation $F_(p, p), F_(q, q), G_(p, p)$ and $C_(p, p)$ respectively, so we have:

$$f * F_(q, q) \equiv (I, I)\ mod\ (q, q) \tag{3.1}$$

$$g * G_(p, p) \equiv (I, I)\ mod\ (p, p) \tag{3.2}$$

$$G_(q, q) * g \equiv (I, I)\ mod\ (q, q) \tag{3.3}$$

$$C_(p, p) * c \equiv (I, I)\ mod\ (p, p) \tag{3.4}$$

Bob next computes the pair of matrices $(h, H) \in \mathbf{M}$ as follows and publish them as his public key.

$$h \equiv w * G_(q, q)\ mod\ (q, q) \tag{3.5}$$

$$H \equiv F_(q, q) * c\ mod\ (q, q) \tag{3.6}$$

He retain $(f, g, c)$ as his private key.

### 3.3. Encryption

Alice chooses her message $m \in L_m$. Next, Alice randomly choose an ephemeral key $r \in L_r$. She encrypts the message as:

$$e \equiv pr * h + H * m\ mod\ (q, q) \tag{3.7}$$

Then she sends $e$ to Bob. Another random $r$ is chosen for next plaintext $m$.

### 3.4. Decryption

Bob for decrypting, computes:

$$\begin{aligned} a &\equiv f * e * g\ mod\ (q, q) \\ &\equiv f * (pr * h + H * m) * g\ mod\ (q, q) \\ &\equiv f * pr * h * g + f * H * m * g\ mod\ (q, q) \\ &\equiv pf * r * w * G_(q, q) * g + f * F_(q, q) * c * m * g \\ &mod\ (q, q) \\ &\equiv pf * r * w + c * m * g\ mod\ (q, q) \end{aligned} \tag{3.8}$$

If $a$ is equal to the non-modular expression $pf*r*w+c*m*g$, Bob can compute the matrices b:

$$\begin{aligned} b &\equiv a\ mod\ (p, p) \\ &\equiv c * m * g\ mod\ (p, p) \end{aligned} \tag{3.9}$$

Hence, the other private keys $C_(p, p)$ and $G_(p, p)$ are used to obtain the plaintext $m$ as:

$$\begin{aligned} m' &\equiv C_(p, p) * c * m * g * G_(p, p)\ mod\ (p, p) \\ &\equiv m\ mod\ (p, p) \end{aligned} \tag{3.10}$$

### 3.5. Decryption Failure

There are some requirements for true parameter selection that we reflect them here. $f*r*w$ and $c*m*g$ have short norms for preventing decryption failure. The key point for decryption without failure is $f*r*w$ and $c*m*g$ are not too large so we want to keep $|pf * r * w + c * m * g|_\infty$ short. For high levels of security, $w$ stays hidden from adversary. Approximately, $|w| \approx |m|$, hence we have $|pf * r * w| \approx |c * m * g|$.

## 4. Security Analysis

In this section, we analyze PairTRU security level for appropriate parameters.

### 4.1. Brute Force Attack

To conduct a brute force technique against PairTRU, attackers who know the public parameters and public key $h \equiv w * G_(q, q)\ mod\ (q, q)$ and $H \equiv F_(q, q) * c\ mod\ (q, q)$ and also, $n, k, p, q$ and $d$. Adversary wants to find the private key $(f, g, c)$ then (s)he can try all possible key in $f, g \in L_f$ so that $h * g$ mod $(q, q)$, $f * H$ mod $(q, q)$ and $c$ mod $(q, q)$ repeatedly for finding decryption flaws. Therefore, adversary wants to find pair of $(f, g)$ that $f$ and $g$ are determined by $2k^2$ polynomials. So the size of the key space $L_f$ is performed as follows:

$$\#L_f = [\frac{n!}{(n - 2d_f)!d_f!^2}]^{2k^2} \tag{4.1}$$

Here $d_f$ and $d_r$ are defined by assuming $L_f$ and $L_r$ contains polynomials from the set of polynomials $L(d_f, d_f)$ and $L(d_r, d_r)$ respectively. Note that just like NTRU, $f, g$ and all of their scalar rotations $(x^i.f, x^i.g)$ can be served as decryption key. Using Meet-In-The-Middle attack [22] the search time could be reduced to $\sqrt{\#L_f/nk^2}$ if enough memory is provided. Since the total state space which an attacker has to search for an encryption key is about $\#L_f/nk^2$. Similarly, the same attack can also be done against a given message by testing all possible $r \in L_r$ and search for the matrices $e - r * h$ mod $(q, q)$ which contains polynomials with small entries. Thus, the message security is $\#L_r/nk^2$ for brute force attack and $\sqrt{\#L_r/nk^2}$ for Meet-In-The-Middle attack, where:

$$\#L_r = [\frac{n!}{(n - 2d_r)!d_r!^2}]^{2k^2} \tag{4.2}$$

For appropriate value of $nk^2$, the brute-force attack is not done on the PairTRU cryptosystem. Meet-In-The-Middle attack cannot be operated on PairTRU because computations involved in decryption are non-commutative.

### 4.2. Chosen Ciphertext Attacks

Because of similarity among PairTRU and NTRU, the security and survivability of the our proposed cryptosystem against adaptively chosen ciphertext attacks [23] is exactly equivalent to NTRU, then one can use prevention techniques [24] for PairTRU. Hence, the PairTRU is CCA-secure.

### 4.3. Message Expansion

In PairTRU the length of the encrypted message is twice as long as the NTRU and is more than the original message and that is costly part for speed in PairTRU cryptosystem. The massage expansion is shown by $2log|E|/2log|P| = logq/logp$, where $E$ is the dictionary of encrypted message and $P$ is the dictionary of plaintext; for NTRU and PairTRU.

### 4.4. Multiple Transmission Attack

Here, utilizing identical public key for message $m$ several times with different error values r's, it is then possible to obtain information on the r's. Suppose Alice transmit different encrypted massages $e_i \equiv r_i * h + H * m$ mod $(q, q)$, then attacker can compute $(e_i - e_1) * h$ mod $(q, q)$. Therefore recovering $r_i - r_1$ mod $(q, q)$ by repeating this operation with the different $e_i$, attacker obtains $r_1$ to provide a brute force attack on the remaining coordinates. Therefore, multiple transmissions is an important security issue.

### 4.5. Analyzing Lattice Attack against the PairTRU

Shamir in [6] showed if we design a variant of NTRU with non-commutative encryption and decryption then the system will be secure against Lattice attacks. In this paper our direction involves extension of the NTRU to broader algebra and non-commutative algebra together for obtaining robust security against linear algebra attack. In this section we prove that the security of PairTRU relies on the intractability of the shortest pair of vectors problem (SPVP).

We can attack this cryptosystem if we find a suitable key for decryption by expanding public key pair $(h, H)$ in which vector $(f * w, c * g)$ lies as a system of linear equations and form a lattice of dimension $2nk^2 \times 2nk^2$. In other words, we show vectors $f * w$ and $c * g$ to be same linear transformation of public key vectors for attack. In the following theorem we prove the security of the PairTRU based on intractability of SPVP in a lattice and some non-linear equations.

**Theorem 4.1.** *Let $(h, H) \subset \mathbf{M}$, and assume that there is a transformation $\rho_{f,g}$ has at least a pair of solutions $f * w$ and $c * g$ in $\mathbf{M}$, then an attacker cannot make a lattice by $h$ and $H$, which contains the vectors $(f * w, c * g)$.*

*Proof:* We know by left multiplying $f$ and right multiplying $g$ to encrypted message, $f * w$ and $c * g$ are produced. We can define the linear map as follows:

$$\begin{aligned} \rho_{f,g} : \mathbf{M} &\longrightarrow \mathbf{M} \\ h &\longrightarrow f * h * g \ \ or \ \ (h \longrightarrow f * w) \qquad (4.3) \\ H &\longrightarrow f * H * g \ \ or \ \ (H \longrightarrow c * g) \qquad (4.4) \end{aligned}$$

The private key $(f*w, c*g)$ viewed as a vector of length $2nk^2$ over $(\mathbb{Z}\times\mathbb{Z})[x]$ belongs to the lattice $L_{PairTRU}$ of dimension and rank $2nk^2$. Let the cyclic shift of the coefficients of the matrices $(h, H)$ gives basis vectors. The lattice $L_{PairTRU}$ is the $(\mathbb{Z}\times\mathbb{Z})[x]$ span of the rows of the matrix $M_{PairTRU}$ defined as:

$$M_{PairTRU} = \begin{bmatrix} [I, I]_{2nk^2\times 2nk^2} & [h, H]_{2nk^2\times 2nk^2} \\ [0, 0]_{2nk^2\times 2nk^2} & [qI, qI]_{2nk^2\times 2nk^2} \end{bmatrix} \qquad (4.5)$$

Clearly, linear transformation in equations 4.3 and 4.4 provides a lattice attack if and only if public key $(h, H)$ constructs a lattice with vector $(f*w, c*g)$ or if following transformation is linear:

$$(h, H) \longrightarrow (f * w, c * g) \qquad (4.6)$$

We show in following analysis that transformation $h \longrightarrow f*h*g$ is not linear. Similarly, one can prove $H \longrightarrow f*H*g$ and $(h, H) \longrightarrow (f * w, c * g)$ are not linear. Consider the multiplication of the matrices $f * h * g = f * w$, where each matrix $(f, g, h, f * w)$ having $k^2$ short polynomials with pairwise coefficients as elements:

$$\begin{bmatrix} f_1 & \cdots & f_k \\ \vdots & \ddots & \vdots \\ f_{k(k-1)} & \cdots & f_{k^2} \end{bmatrix} \cdot \begin{bmatrix} h_1 & \cdots & h_k \\ \vdots & \ddots & \vdots \\ h_{k(k-1)} & \cdots & h_{k^2} \end{bmatrix}$$
$$\cdot \begin{bmatrix} g_1 & \cdots & g_k \\ \vdots & \ddots & \vdots \\ g_{k(k-1)} & \cdots & g_{k^2} \end{bmatrix}$$
$$= \begin{bmatrix} f * w_{1,1} & \cdots & f * w_{1,k} \\ \vdots & \ddots & \vdots \\ f * w_{k,1} & \cdots & f * w_{k,k} \end{bmatrix} \qquad (4.7)$$

Now we can show system of equations as follows:

$$\begin{aligned} & g_1 f_1 h_1 + ... + g_{k(k-1)+1} f_1 h_k + ... + g_{k(k-1)+1} f_k h_{k^2} \\ & \qquad = (fw)_{1,1} \\ & g_2 f_1 h_1 + ... + g_{k(k-1)+2} f_1 h_k + ... + g_{k(k-1)+2} f_k h_{k^2} \\ & \qquad = (fw)_{1,2} \\ & \qquad \vdots \\ & g_k h_1 f_{k(k-1)+1} + ... + g_{k^2} h_k f_{k(k-1)+1} + ... + g_{k^2} h_{k^2} f_{k^2} \\ & \qquad = (fw)_{k,k} \end{aligned} \qquad (4.8)$$

So general term can be represented as:

$$(fw)_{i,j} = \sum_{m=k(i-1)+1}^{ki} \sum_{s=0}^{k-1} f_m (g_{j+sk})(h_{(1+s)(m-k(i-1))}) \qquad (4.9)$$

Or, another form is:

$$(fw)_{i,j} = \sum f_l g_m h_z = \sum U_z h_z \qquad (4.10)$$

where $i, j, l, m \in [1, k^2]$; $z \in [1, k^4]$.

As all $U_z$ are not identical so we cannot obtain $X_i = (x_1, x_2, ..., x_{k^2})$ so that gives $f * w$ by multiplying with a lattice using cyclic shift of the coefficients of $h$. In nutshell, we cannot find different vector $X_i$ to multiply $M_{PairTRU}(V)$ with $v_1, v_2, ..., v_{2nk^2}$ to get $f * w$ as a short lattice vector. We therefore conclude:

$$f * w \neq X_i L_{PairTRU}(v_1, v_2, ..., v_{2nk^2}) \qquad (4.11)$$

□

Therefore, an attacker cannot produce a lattice by $h$ and $H$, which contains the vectors $(f * w, c * g)$. So lattice attack will not work for the PairTRU cryptosystem. Also in practice, for dimension 150 and upper, brute force search is inefficient [25], [26]. Therefore, attacker can use polynomial-time methods such as LLL or system of equation attack. In PairTRU, we change the ring $\mathbb{Z}$ in the NTRU by $\mathbb{Z}\times\mathbb{Z}$. Notice that the PairTRU like the NTRU is resistant against LLL-like methods or CRT which is the important threat to the original NTRUEncrypt. Clearly, NTRU has 2N-dimensional lattice and lattice attacks are not effective for $N = 251$. Hence, lattice dimension or key size is reduced approximately by factor of 2. In this paper, as we see, the PairTRU cryptosystem has 4N-dimensional lattice therefore it's key size is reduced

TABLE 2
Coordinatewise comparison between the PairTRU with the NTRU.

| Characteristics | NTRU | PairTRU |
|---|---|---|
| Plaintext Block | $Nlog_2p$ (bits) | $(nk^2log_2p, nk^2log_2p)$ (bits) |
| Ciphertext Block | $Nlog_2q$ (bits) | $(nk^2log_2q, nk^2log_2q)$ (bits) |
| Encryption Speed | $O(N^2)$ | $(\tilde{O}(nlogn.k^{2.376}), \tilde{O}(nlogn.k^{2.376}))$ |
| Decryption Speed | $O(N^2)$ | $(\tilde{O}(nlogn.k^{2.376}), \tilde{O}(nlogn.k^{2.376}))$ |
| Message Expansion | $log_pq$ to 1 | $(log_pq$ to $1, log_pq$ to $1)$ |
| Private Key Length | $2Nlog_2p$ (bits) | $(2nk^2log_2p, 2nk^2log_2p)$ (bits) |
| Public Key Length | $Nlog_2q$ (bits) | $(2nk^2log_2q, 2nk^2log_2q)$ (bits) |
| Lattice Security | Secure in | Secure in |
| | High Dimension | Low Dimension |

approximately by factor of 4. In nutshell, we could construct an efficient NTRU-Like public key cryptosystem with reliable and comparable lattice security.

## 5. Performance Analysis and Comparison with the NTRU

In this section, comparison of the PairTRU and the NTRU is described and is shown in Table 2. The PairTRU cryptosystem utilizes five positive integer $(n, k, p, q, d)$ with $p$ and $q$ coprime and five sets of matrices $(L_f, L_c, L_r, L_w, L_m) \subset \mathbf{M}$. The properties are listed in terms of the parameters $(N, p, q)$ for the NTRU. We give comparison by choosing $N = nk^2$, since this equates to plaintext message blocks of the same size but with pairwise coefficient. In table 2, since the PairTRU calculates left and right multiplication for decrypting, so the fix element will be about four times higher than of the NTRU. For LLL attack, the PairTRU needs private key and public keys with matrix form which are different from the NTRU private key and public key. However, the PairTRU gives efficient speed over the NTRU. Note that we can reduce the number of encryption and decryption operations to $\tilde{O}((nlogn)k^{2.376})$, if we use FFT for polynomial multiplication, which is considerable speed improvement over the NTRU.

## 6. Conclusion

We studied the NTRU over ring $R = \frac{\mathbb{Z}[x]}{<x^N-1>}$ but we found that in low dimension the NTRU is less insecure against Lattice-based attacks and linear algebra-based attacks. We extend this system over non-commutative matrix quotient polynomial ring

$$\mathbf{M} = \frac{M(k, \mathbb{Z} \times \mathbb{Z})[x]}{< (I_{k\times k}, I_{k\times k})x^N - (I_{k\times k}, I_{k\times k}) >},$$

where $\mathbf{M}$ is a matrix ring of $k \times k$ matrices of polynomials in $R = \frac{(\mathbb{Z}\times\mathbb{Z})[x]}{<(1,1)x^N-(1,1)>}$. The PairTRU security level is comparable to the NTRU with respect to several well-known attacks with significant speed improvement. Also, we have shown that the PairTRU cryptosystem is more secure than the NTRU, because of its lattice structure and robustness against linear algebra attack. Here, security and efficiency of non-commutative PairTRU cryptosystem are proved. In the end, we would like to point out that the PairTRU is the first step in extension of the NTRU public-key cryptosystems with pairwise coefficient in quotient polynomial rings. Furthermore, the PairTRU can be generalized to different types of rings, modules, and vector spaces, or different kinds of algebras in order to design new lattice-based cryptosystems and explore their possible advantages.

## References

[1] R.A. Perlner, and D.A. Cooper, *Quantum resistant public key cryptography: a survey*, In: Proc. of IDtrust, ACM, New York, 2009, pp. 85–93.

[2] J. Hoffstein, J. Pipher, and J.H. Silverman, *NTRU: A Ring-Based Public Key Cryptosystem*, Algorithmic Number Theory (ANTS III), Portland, OR, June 1998, J.P. Buhler (ed.), LNCS 1423, Springer-Verlag, Berlin, 1998, pp. 267–288.

[3] J. Hoffstein, J.H. Silverman, and W. Whyte, *Estimated Breaking Times for NTRU Lattices*, Technical Report #12, available at www.ntru.com.

[4] I. Blake, G. Seroussi, and N. Smart, *Elliptic Curves in Cryptography*, Cambridge University Press, Cambridge, 1999.

[5] http://www.ntru.com.

[6] D. Coppersmith, and A. Shamir, *Lattice attacks on NTRU*, in EUROCRYPT ‘97, 1997, pp. 52–61.

[7] C. Gentry, *Key recovery and message attacks on NTRU-composite*, In Eurocrypt ‘01, Springer LNCS 2045, 2001, pp. 182–194.

[8] Standard Specifications for Public-Key Cryptographic Techniques Based on Hard Problems over Lattices. IEEE P1363, 2008. Available at http://grouper.ieee.org/groups/1363/.

[9] D. Han, J. Hong, J.W. Han, and D. Kwon, *Key recovery attacks on NTRU without ciphertext validation routine*, In Proceeding of ACISP ‘03, LNCS, Springer-Verlag, vol. 2727, 2003, pp.274–284.

[10] E. Jaulmes, and A. Joux, *A Chosen Ciphertext Attack on NTRU*, In Proceeding of CRYPTO ‘00, LNCS, Springer-Verlag, vol. 1880, 2000, pp. 20–35.

[11] N. Howgrave-Graham, P.Q. Nguyen, D. Pointcheval, J. Proos, J.H. Silverman, A. Singer, and W. Whyte, *The Impact of Decryption Failures on the Security of NTRU Encryption*, In Proceeding of CRYPTO ‘03, LNCS, Springer-Verlag, vol. 2729, 2003, pp. 226–246.

[12] P.Q. Nguyen, and D. Pointcheval, *Analysis and Improvements of NTRU Encryption Paddings*, In Proceeding of CRYPTO ‘02, LNCS, Springer-Verlag, vol. 2442, 2002, pp. 210–225.

[13] P. Gaborit, J. Ohler, and P. Sole, *CTRU, a polynomial analogue of NTRU*, Tech- nical report, INRIA, France, 2002. Available at ftp://ftp.inria.fr/INRIA/publication/ publi-pdf/RR/RR-4621.pdf.

[14] M. Coglianese, and B.M. Goi, *MaTRU: A New NTRU-Based Cryptosystem*, In Proceedings of the 6th International Conference on Cryptology in India (INDOCRYPT), 2005, pp. 232–243.

[15] N. Vats, *NNRU, a Noncommutative Analogue of NTRU*, The Computing Research Repos- itory (CoRR), abs/0902.1891, 2009. Available at http://arxiv.org/abs/0902.1891.

[16] R. Kouzmenko, *Generalizations of the NTRU Cryptosystem*, Master’s thesis, Polytechnique Montreal, Canada, 2006.

[17] C. Karimianpour, *Lattice-Based Cryptosystems*, Master’s thesis, University of Ottawa, Canada, 2007.

[18] M. Nevins, C. Karimianpour, and A. Miri, *NTRU over rings beyond $\mathbb{Z}$*, Designs, Codes and Cryptography, vol. 56, no. 1, 2010, pp. 65–78.

[19] E. Malekian, A. Zakerolhosseini, and A. Mashatan, *QTRU: Quaternionic Version of the NTRU Public-Key Cryptosystems*, The int’l Journal of information Security (ISeCure), vol. 3, no. 1, 2011, pp. 29–42.

[20] A.H. Karbasi and R.E. Atani, *ILTRU: An NTRU-Like Public Key Cryptosystem Over Ideal Lattices*, IACR Cryptology ePrint Archive 2015: 549, 2015.

[21] M.P. Karampetakis, and P. Tzekis, *On computation of the genralized inverse of a polynomial matrix*, IMA, vol. 18, 2001, pp. 83–97.

[22] N. Howgrave-Graham, J.H. Silverman, and W. Whyte, *A Meet-In-The-Middle Attack on an NTRU Private Key*, Technical report, Security Innovation Inc., Boston, MA, USA, 2002. Available at http://securityinnovation.com/cryptolab/pdf/NTRUTech004v2.pdf.

[23] E. Jaulmes, and A. Joux, *A Chosen Ciphertext Attack against NTRU*, In Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO ‘00), 2000, pp. 20–36.

[24] J. Hoffstein, and J.H. Silverman, *Optimizations for NTRU*, Technical Report 015, NTRU Cryptosystems, 2000. Available at http://www.sisecure.com/cryptolab/pdf/ TECH-ARTICLE-OPT.pdf.

[25] P.Q. Nguyen, and D. Stehlé, *LLL on the Average*, In Proceedings of the 7th International Symposium on Algorithmic Number Theory (ANTS-VII )., 2006, pp. 238–256.

[26] P.Q. Nguyen, and D. Stehlé, *Low Dimensional Lattice Basis Reduction Revisited*, ACM Transactions on Algorithms, vol. 5, no. 4, 2009, pp.1–48.

[27] A.H. Karbasi and R.E. Atani, *PSTRU: A provably secure variant of NTRUEncrypt over extended ideal lattices*, The 2nd National Industrial Mathematics Conference, Tabriz, Iran, 2015.

[28] A.H. Karbasi and R.E. Atani, *A Survey on Lattice-based Cryptography*, (In Persian), Biannual Journal for Cyberspace Security (Monadi AFTA), Vol. 3, No. 1, 2015, pp 3–14. Available from: http://monadi.isc.org.ir/browse.php?a_id=23&sid=1&slc_lang=en

[29] S.E. Atani, R.E. Atani, and A.H. Karbasi, *NETRU: A Non-Commutative and Secure Variant of CTRU Cryptosystem*, The ISC international journal of information security (IseCure), to appear.

[30] S.E. Atani, R.E. Atani, and A.H. Karbasi, *EEH: A GGH-Like Public Key Cryptosystem Over The Eisenstein Integers Using Polynomial Representations*, The ISC international journal of information security (IseCure), Vol 7, No. 2, 2015, pp. 115–126.

[31] A.H. Karbasi, R.E. Atani, and S.E. Atani, *A New Ring-Based SPHF and PAKE Protocol On Ideal Lattices*, Submitted.

[32] A.H. Karbasi, M.A. Nia, and R.E. Atani, *Designing of An Anonymous Communication System Using Lattice-based Cryptography*, Journal of Electronic and Cyber Defence, Vol. 2, No. 3, 2014, pp. 13–22, Persian.

[33] S.E. Atani, R.E. Atani, and A.H. Karbasi, *A Provably Secure Variant of ETRU Based on Extended Ideal Lattices over Direct Product of Dedekind domains*, Submitted.

[34] S. Singh and P. Sahadeo, *Generalisations of NTRU cryptosystem*, Security and Communication Networks, DOI: 10.1002/sec.1693, 2016.