DTRU1: First Generalization of NTRU
Using Dual Integers

Mamadou Ghouraïssiou Camara, Demba Sow and Djiby Sow

Laboratoire d’Algèbre, de Cryptologie, de géométrie Algébrique et Applications
Ecole Doctorale de Mathématiques et Informatique
Université Cheikh Anta Diop, Dakar Sénégal

Copyright © 2018 Mamadou Ghouraïssiou Camara, Demba Sow and Djiby Sow. This article is distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

NTRU is the first public key Cryptosystem based on the polynomial ring \( \mathbb{Z}[X]/(X^N - 1) \). The hard problem underlying this cryptosystem is related to finding short vectors in a lattice.

Several generalizations of NTRU was designed over various integral ring such as \( \mathbb{Z}, \mathbb{Z}[i], \mathbb{Z}[w] \) and \( \mathbb{H} \).

In this paper, we use the ring with zeros divisors \( \mathbb{D} = \mathbb{Z} + \epsilon \mathbb{Z}, \epsilon^2 = 0 \) (called the ring of Dual Integers) in order to design a new version of NTRU. To achieve this objective, we have studied the elementary arithmetic properties of the ring of Dual integers in a previous paper. The main difficulty is to be able to perform a division algorithm with a unique remainder and to invert polynomials with coefficients in quotient ring of the ring of Dual integers. Nevertheless, we have successfully design NTRU over Dual integers (called DTRU) in a particular quotient ring of the ring of Dual integers. Our scheme has the same level security than NTRU, but is not more efficient. This work shows also that NTRU can be designed even if the ring has zeros divisors!

We have also design over the ring of Dual Integer the cryptosystem NTRU with Non-inverible polynomial proposed by Banks and Shparlinski. This this version is more secure than NTRU but is less efficient too.

Keywords: Public key Cryptography, Dual integer, pseudo-norm, pseudo-division, Cryptosystem, NTRU
1. Introduction

NTRU is a public key Cryptosystem proposed in 1996 by J. Hoffstein, J. Pipher, and J. H. Silverman in [9]. It is a fast public key cryptosystem based on the polynomial ring \( \mathbb{Z}[X]/(X^N - 1) \). The hard problem underlying this cryptosystem is related to finding short vectors in a lattice (see [5], [7], [19], [15]) due to the properties of polynomials with "small coefficients" used in the Key Generation algorithm and the Encryption algorithm of NTRU Cryptosystem. NTRU is subject to some attack but the more dangerous attack are "Middle man attack" and "Lattice attack" on the public key and the cipher text. But a suitable choice of the size of the parameters allows to prevent theses attacks. A correct choice of padding for NTRU encryption called NEAP was proposed in [12] with a security proof in the presence of description failure. In 2005, an algorithm was designed by Howgrave \textit{et al.} [13], for a good choice of the size of the parameters relatively to the security parameters. Since descriptions failures are a lack for NTRU and all it's variants, Dwork \textit{et al.} have designed in 2007 a method for "Immunizing Encryption Schemes from Decryptions errors" [21].

Many authors have already generalized NTRU-encryption in different ways over rings.

We have the following generalizations.
- In 2000, William D. Banks and Igor E. Shparlinski have proposed in [1] a new a variant of NTRU with Non-Invertible Polynomials on the same ring as NTRU. This generalization is more secure against some of the known attacks on classical NTRU such as lattice attack. But this NTRU is less efficient than the classical NTRU scheme because the length of public key and the cipher text are roughly doubled.
- In 2005, Michael Coglianese and Bok-Min Goi have proposed MaTRU, a new NTRU Based matrices with coefficients in the same ring as NTRU. The cryptosystem MaTRU uses a more efficient linear transformation and provide a security level comparable to that of NTRU.
- In 2006, Gaborit, Ohler and Solé have proposed CTRU where the ring \( \mathbb{Z} \) in NTRU is replaced by \( \mathbb{A} = \mathbb{F}_2[T] \). CTRU was broken by Kouzmenko in his Phd Diploma [14] in 2006 and by Vats [22] in 2008.
- In 2006, Kouzmenko in his Phd Diploma in [14] have proposed a new variant where the ring \( \mathbb{Z} \) in NTRU is replaced by the ring of Gaussian Integer \( \mathbb{Z} + i\mathbb{Z} = \{a + ib \mid a, b \in \mathbb{Z}, \quad i^2 = -1\} \). This scheme is slightly more secure relatively to lattice attack than NTRU but it is not as efficient as NTRU.
- In 2009, Nevins, KarimianPour and Miri [18] have proposed a new variant where the ring \( \mathbb{Z} \) in NTRU is replaced by the ring of Einstein Integers \( \mathbb{Z} + w\mathbb{Z} = \{a + wb \mid a, b \in \mathbb{Z}, \quad i^2 = -1, \quad w = e^{2\pi i/3}\} \). This scheme is similar to NTRU over Gaussian integers.
- In 2009, Malekian, Zakerolhosseini et Mashatan [17] have proposed a new variant where the ring \( \mathbb{Z} \) in NTRU is replaced by the ring of quaternions...
\( \mathbb{H} = \{ a + ib + jc + kd \mid a, b, c, d \in \mathbb{Z}, \; i^2 = j^2 = k^2 = ijk = -1 \} \). This scheme is similar to NTRU over Gaussian integers or Einstein integers.

All these above generalizations of NTRU work over an integral ring such as \( \mathbb{Z} \), \( \mathbb{Z}[i] \), \( \mathbb{Z}[w] \) and \( \mathbb{H} \).

In this paper, we use the ring with zeros divisors \( \mathbb{Z} + \epsilon \mathbb{Z} \), \( \epsilon^2 = 0 \) (called the ring of Dual Integers) in order to design a new version of NTRU.

To achieve this objective, we have studied the elementary arithmetic properties of the ring of Dual integers in a previous paper [2].

The main difficulty is to be able to perform a division algorithm with a unique remainder and to invert polynomials with coefficients in ring of Dual integers.

Nevertheless, we have successfully design NTRU over Dual integers (called DTRU) in a particular quotient ring of the ring of Dual integers. Our scheme has the same level security than NTRU, but is not more efficient. This work shows also that NTRU can be designed even if the ring has zeros divisors!

We have also design over the ring of Dual Integer the cryptosystem ”NTRU with non-inverible polynomial” proposed by Banks and Shparlinski [1]. This this version is more secure than NTRU but is less efficient too.

This paper is organized as follows.

- **Section 1**: NTRU over \( \mathbb{Z} \) and NTRU over Gaussian integer \( \mathbb{Z} + i\mathbb{Z} \)
- **Section 3**: Arithmetic on the ring of Dual integer \( \mathbb{D} = \mathbb{Z} + \epsilon\mathbb{Z} \)
- **Section 4**: DTRU1: NTRU over Dual integer \( \mathbb{D} = \mathbb{Z} + \epsilon\mathbb{Z} \)
- **Section 4**: DTRU2: NTRU with Non-inverible polynomial over Dual integer \( \mathbb{D} = \mathbb{Z} + \epsilon\mathbb{Z} \)

2. NTRU Cryptosystem \( \mathbb{Z} \) and NTRU over Gaussian integer \( \mathbb{Z} + i\mathbb{Z} \)

In this section we recall the properties of the classical NTRU Cryptosystem that we want to generalize in this paper.

2.1. NTRU encryption scheme.


The original NTRU cryptosystem depends on three parameters \((N, p, q)\) (with \(N\) an integer) and four set \( \mathcal{L}_f, \mathcal{L}_g, \mathcal{L}_\alpha, \mathcal{L}_m \) of polynomials of degree \( N - 1 \) with integer coefficients. Note that \(p\) and \(q\) need not be prime, but we will assume that \(gcd(p, q) = 1\), \(N\) is prime and \(q\) will always larger than \(p\). Let \( R = \frac{\mathbb{Z}[x]}{(x^n - 1)} \), an element \( F \in R \) will be written as follow:

\[
F = \sum_{i=0}^{N-1} F_i x^i = (F_0, F_1, ..., F_{N-1})
\]
The multiplication of two polynomials $F$ and $G$ is defined as the cyclic convolution of their coefficients: $H = F \ast G$ with

$$H_k = \sum_{i+j=k \pmod{N}} F_i G_j = \sum_{i=0}^{k} F_i G_{k-i} + \sum_{i=k+1}^{N-1} F_i G_{N+k-i}$$

which is the ordinary polynomial multiplication modulo $X^N - 1$.

When we do a multiplication modulo $q$ we mean to reduce the coefficients modulo $q$.

**Key Generation**

1. Chosen at random a polynomial $f \in \mathcal{L}_f$, invertible $\mod q$ and $\mod p$, and denote these inverses by $f_q$ and $f_p$, that is, $f_q \ast f \equiv 1 \pmod{q}$ and $f_p \ast f \equiv 1 \pmod{p}$.
2. Chosen at random a polynomial $g \in \mathcal{L}_g$.
3. Compute the polynomial $h = p \cdot f_q \ast g \pmod{q}$, $h$ is the public key.

**NB:** The polynomial $f$ and $g$ generally have small coefficients, while $h$ has large coefficients.

**Encryption** Suppose that Aissa wants to send a message to Oumar.

1. She begins by selecting a message $m$ from the set of plaintexts $\mathcal{L}_m$.
2. Next, she randomly chooses a polynomial $\phi \in \mathcal{L}_\phi$.
3. and uses the Oumar’s public key $h$ to compute $e = \phi \ast h + m \pmod{q}$
4. $e$ is the encrypted message which Aissa sends to Oumar.

**Decryption** Why decryption works?

Suppose that Oumar has received the message $e$ transmitted by Aissa and wants to decrypt it using his private key $f$. To do this efficiently, Oumar should have precomputed the polynomial $f_p$.

1. In order to decrypt $e$, Oumar, first computes: $a = f \ast e \pmod{q}$
2. Chooses the coefficients of $a$ in the interval from $-\frac{q}{2}$ to $\frac{q}{2}$.
3. Omar recovers the message by computing $f_p \ast a \pmod{p}$.

**Security of NTRU**

The most practical attacks against NTRU are:

- the Lattice attacks on NTRU public key and NTRU cipher text,
- the Meet-in-the-Middle attack on NTRU public key and NTRU cipher text,
- the Hybrid lattice-reduction and Meet-in-the-Middle Attack Against NTRU,
- Attacks using decryptions failures,
- and the Chosen cipher texts attacks.

To circumvent these attacks, a new version of NTRU encryption called NEAP was proposed in [12] with a security proof in the presence of description failure. After, an algorithm was designed by Howgrave et al. [13], for a good choice of the size of the parameters relatively to the security parameter. A method for ”Immunizing Encryption Schemes from Decryptions errors” was
proposed by Dwork et al. and this technic allow to solve the probleme of descriptions failures for NTRU.

2.2. GTRU: NTRU over Gaussian Integer $\mathbb{Z} + i\mathbb{Z}$.

NTRU was first generalized on a new ring, (namely the ring of Gaussian integers) by R. Kouzmenko in his diploma project in 2006.

In this section, we describe the encryption/decryption scheme of NTRU using the ring of Gaussian integers (see[14]). For arithmetic properties of Gaussian integers and the definition of associated the norm $d$, we will follow (see[14])

Recall that if $z = a + ib \in \mathbb{Z}[i]$ (with $i^2 = -1$), then:

$$d(z) = a^2 + b^2,$$

and $|f|_{\infty} = \max_i \{d(f_i)\}$ if $f = (f_0, \ldots, f_{N-1}) \in \mathbb{Z}[i][x]$.

Let $p, q \in \mathbb{Z}[i]$ be to Gaussian integers. Let $f, g$ be polynomials in $\mathbb{Z}[i][x]$. Suppose that $f$ is invertible in $\mathbb{Z}[i]/(p)[x]$ and $\mathbb{Z}[i]/(q)[x]$ with inverses respectively $f_p$ and $f_q$.

The public key $h$ is define as follow:

$$h = pg * f_q \in \mathbb{Z}[i]/(q)[x]$$

Let $m \in \mathbb{Z}[i]/(p)[x]$ such that $|m|_{\infty} < \frac{d(p)}{4}$. Let $r \in \mathbb{Z}[i]/(q)[x]$. The cipher text is:

$$e := r * h + m \in \mathbb{Z}[i]/(q)[x]$$

Suppose that

$$|pr * g + f * m|_{\infty} < \frac{d(q)}{4}$$

where the operations to compute the polynomial are done in $\mathbb{Z}[i][x]$. Then, if we multiply $e$ by $f$, reduce all coefficients mod $q$, then mod $p$ and multiply the result by $f_p$, we recover $m$.

This scheme designed by Kouzmenko over the ring of Gaussian integers, is slightly more secure relatively to lattice attack than NTRU but it is not as efficient as NTRU.

3. Arithmetic on Dual Integer $\mathbb{D} = \mathbb{Z} + \epsilon\mathbb{Z}$ with $\epsilon^2 = 0$

In this section we recall the arithmetic properties on dual integer $\mathbb{D} = \mathbb{Z} + \epsilon\mathbb{Z}$ (with $\epsilon^2 = 0$), in particular the algorithm that allow us to reverse a polynomial in a modular ring of $\mathbb{D}[x]$.

3.1. The Ring of Dual integers.

Let us recall (see[2]) the definition and the elementary arithmetics of the Dual integers.

**Definition 3.1.** We denote $\mathbb{D} = \mathbb{Z}[\epsilon]$ with $\epsilon^2 = 0$. The ring $\mathbb{D}$ is called the ring of dual integers. If $z = a + \epsilon b \in \mathbb{D}$ then $a$ is called the real part of $z$ and is denoted $Re(z)$; $b$ is called the imaginary part of $z$ and is denoted $Im(z)$. 
Theorem 3.2. (see [2])

$(\mathbb{D}, \varphi)$ is a pseudo-euclidian ring. Indeed if $z, t \in \mathbb{D}$ and $t$ a non zero divisor, so we can find $(q, r) \in \mathbb{D}^2$ such as $z = tq + r$
with $r = 0$ or $\varphi(r) < \frac{\varphi(t)}{4}$ (remark that $(q, r)$ is not necessary unique).

Algorithm of pseudo-division

Input: $z \in \mathbb{D}$ and $t \in \mathbb{D}\setminus J_\mathbb{D}$;
Output: $(q, \rho) \in \mathbb{D}$ such that $z = qt + r$ with $\varphi(\rho) < \frac{\varphi(t)}{4}$.

1. Put $a_1 \leftarrow \text{Re}(zt)$, $a_2 \leftarrow \text{Im}(zt)$ and $n = t \bar{t}$
2. Find by division algorithm $(q_i, r_i)$ with $i = 1, 2$ such that $a_i = nq_i + r_i$;
3. output $q = q_1 + q_2 \epsilon$ and $\rho = z - tq$.

Note that the remainder in the pseudo division is not unique but our algorithm output always the same remainder and the smallest one relatively to our pseudo-norm. We are going to prove these two properties in the following two theorems.

Theorem 3.3. (see [2]) The pseudo-division algorithm described in the proof of the theorem 4.2. yields unique representative $\leftrightarrow \mod t$. More precisely, if $z_1, z_2, t$ are three dual integers such that $t$ non zero divisors and

$z_1 \equiv z_2 \mod t$

Then the algorithm outputs two remainders $r_1'$ and $r_2'$ such that $r_1' = r_2'$.

In the following theorem, we prove that our algorithm output the smallest possible remainder relatively to our pseudo norm.

Theorem 3.4. (see [2]) Let $y, t$ be two dual integers. Suppose that $t \in \mathbb{D}\setminus J_\mathbb{D}$. If $d(y) < \frac{1}{4} \varphi(t)$, then all dual integers equal to $y \mod t$ are reduced to $y$ using the algorithm described previously.

Proposition 3.5. (see [2]) Let $y, t$ be two dual integers with $t \in \mathbb{D}\setminus J_\mathbb{D}$. If the pseudo-division algorithm outputs $(q, r)$ such that $y = tq + r$ with $r = 0$ or $\varphi(r) < \frac{1}{4} d(t)$ then for each integer $m$, the dual integers $q' = q + me$ and $r' = r - t_1 me$ (where $t_1 = \text{Re}(t)$) verify $y = tq' + r'$ with $\varphi(r') = \varphi(r) < \frac{1}{4} \varphi(t)$

Remark

4 is the best constant that we find for this be true.
3.2. On the invertibility modulo a polynomial in $\frac{D}{(z)}[x]$.

Let $f, g \in \frac{D}{(z)}[x]$ (with $z \in \mathbb{D}$), we say that $f$ and $g$ are co-prime (denoted $f \wedge g = 1$) if there exists $f', g' \in \frac{D}{(z)}[x]$ such that $ff' + gg' = 1 \mod z$.

Let $p$ be a prime integer (then $p$ is irreducible in $\mathbb{D}$).

Write

$$f = f_1 + \epsilon f_2 \in \frac{D}{(p\mathbb{D})}[x], \quad f_i \in \frac{\mathbb{Z}}{p\mathbb{Z}}[x], \quad i = 1, 2$$

and

$$g = g_1 + \epsilon g_2 \in \frac{D}{(p\mathbb{D})}[x], \quad g_i \in \frac{\mathbb{Z}}{p\mathbb{Z}}[x], \quad i = 1, 2$$

How to compute the inverse of $f \mod g$ in $\frac{D}{(p\mathbb{D})}[x]$?

**Pseudo-extending Euclidean Algorithm**

**Input:** $f = f_1 + \epsilon f_2 \in \frac{D}{(p\mathbb{D})}[x]$, and $g = g_1 + \epsilon g_2 \in \frac{D}{(p\mathbb{D})}[x]$ with $g_i, g_i \in \frac{\mathbb{Z}}{p\mathbb{Z}}[x], \quad i = 1, 2$

1. if $gcd(f_1, g_1) \neq 1$ in $\frac{\mathbb{Z}}{p\mathbb{Z}}[x]$ output "$f$ is not invertible in $\frac{D}{(g(x))}$",
2. if $gcd(f_1, g_1) = 1 \in \frac{\mathbb{Z}}{p\mathbb{Z}}[x]$

   a. compute by extending Euclidean algorithm $(u_1, v_1) \in \frac{\mathbb{Z}}{p\mathbb{Z}}[x]$ such as $f_1 u_1 + g_1 v_1 = 1 \in \frac{\mathbb{Z}}{p\mathbb{Z}}[x]$  

   b. $h \leftarrow -f_2 u_1 - g_2 v_1$, $u_2 \leftarrow u_1 h$ and $v_2 \leftarrow v_1 h$

   c. output $u \leftarrow u_1 + \epsilon u_2 = u_1 + \epsilon u_1 h$ and $v \leftarrow v_1 + \epsilon v_2 = v_1 + \epsilon v_1 h$

**Output:**

- "$f$ is not invertible in $\frac{D}{(g(x))}$" if $gcd(f_1, g_1) \neq 1 \in \frac{\mathbb{Z}}{p\mathbb{Z}}[x]$

- $u = u_1 + u_2 \epsilon \in \frac{D}{(p\mathbb{D})}[x]$ and $v = v_1 + v_2 \epsilon \in \frac{D}{(p\mathbb{D})}[x]$ such that $fu + gv = 1 \in \frac{D}{(p\mathbb{D})}[x]$, if $gcd(f_1, g_1) = 1 \in \frac{\mathbb{Z}}{p\mathbb{Z}}[x]$

**OPEN PROBLEME:** Find a Pseudo-extending Euclidean Algorithm to invert a polynomial $f$, invertible $\frac{D}{(g(x))}$ where $z \in \mathbb{D} \setminus \mathbb{Z}$

4. **DTRU:** NTRU over Dual integer $\mathbb{D} = \mathbb{Z} + \epsilon \mathbb{Z}$

4.1. **NTRU over a Pseudo-Euclidean Rings.** Let us recall the definition of a Pseudo-Euclidean Rings.

A ring $A$ is pseudo-Euclidean, if there exists an application $\varphi : A \rightarrow \mathbb{N}$ verifying the following properties:

1) $\varphi(z) \geq 0 \quad \forall z \in A$

2) $\forall z \in A$ and for all $t \in A - J_A$ where $J_A$ is the set of zero divisors we have:

$$\varphi(z) \leq \varphi(zt)$$

3) Let \( z \in A \) and let \( t \in A - J_A \) (where \( J_A \) is the set of zero divisor) then there exist \((q, r) \in A^2\) such that \( z = tq + r \) with \( r = 0 \) or \( \varphi(r) < \varphi(t) \).

The application \( \varphi \) defined above, is called pseudo-norm.

We briefly outline the encryption/decription scheme using a ring with a zero-divisors \( A \), and explain why it is convenient if \( A \) is Pseudo-Euclidean. Let \( N \) be an integer, \( p \) and \( q \) be two elements of \( A \). Define
\[
R = A[x], \quad \mathcal{R}_p = \frac{A}{\mu A[x]}(x^N - 1), \quad \mathcal{R}_q = \frac{A}{\nu A[x]}(x^N - 1)
\]
Where \((p)\) and \((q)\) denote the ideals generated by respectively \( p \) and \( q \). Let \( L_m, L_q \) be subsets of respectively \( \mathcal{R}_p \) and \( \mathcal{R}_q \), and let \( L_f, L_\phi \) be subsets of \( \mathcal{R} \). We will identify an element of the ring \( \mathcal{R} \) with its projection in \( \mathcal{R}_p \) and \( \mathcal{R}_q \).

**Key generation** To generate keys, pick two elements \( f \in L_f \) and \( g \in L_q \), with \( f \) invertible in \( \mathcal{R}_p \) and \( \mathcal{R}_q \) and denote \( f_p, f_q \) its inverses respectively; compute \( h = f_qg + f \in \mathcal{R}_q \) and output \( h \) the public key.

**Encryption** To encrypt \( m \in L_m \), pick \( \phi \in L_\phi \) and compute \( e = p\phi + h + m \in \mathcal{R}_q \) and output \( e \) as ciphertext.

**Decryption** To decrypt, multiply \( e \) by \( f \) to get \( a = p\phi + f \in \mathcal{R}_q \); from \( a \), deduce the expression \( p\phi + f \in \mathcal{R}_q \) and multiply by \( f_q \) to recover \( m \).

**Discussion:**

As stated by Kouzmenko in his thesis about Euclidean ring, (see[14]), there are also two problems in the above procedure for Pseudo-Euclidean ring, if we want to apply it in practice:

1. How to invert \( f \).
2. How to switch from computation mod \( p \) to computation mod \( q \) in the decryption part.

To meet this criteria, we must have the following:

1. "Pseudo-extending Euclidean Algorithm" for inverting \( f \) in \( \frac{A}{(p)}[x] \) and \( \frac{A}{(q)}[x] \) if it is indeed invertible;
2. define \( |f|_{\infty} = \max_{0 \leq i \leq N-1} \varphi(f_i) \), then choose parameters in such a way that \( |p\phi + f \cdot m|_{\infty} < \varphi(q) \) (E) and switch from computation mod \( p \) to mod \( q \).

Since in general the pseudo-division algorithm does not produce unique ‘remainders’, we need to modify the above procedure(this will be our modification for the Dual integers), there may exist a constant \( K \) such that if \( \varphi(z) < \varphi(q) \) then \( z \) is the unique representative of the equivalence class mod\( q \) with this property and hence we must replace \( \varphi(q) \) by \( \varphi(q) \) in equation (E).

In the last par of this subsection, to illustrate these ideas, we will show how NTRU can be modeled over the ring of Dual Integers.
NTRU over Dual integers

Now, summing up the results of the section 3, we obtain the following Theorem for DTRU over Dual integers.

**Theorem 4.1.** Let \( p, q \in \mathbb{D} \) be dual integers.

1. Let \( f, g \) be polynomials in \( \mathbb{D}[x] \), \( f \) invertible in \( \mathbb{D}(p)[x] \) and \( \mathbb{D}(q)[x] \) with inverses respectively \( f_p \) and \( f_q \).
2. Define \( h = g \ast f_q \in \mathbb{D}(q)[x] \).
3. Let \( m \in \mathbb{D}(p)[x] \) such that \( |m|_\infty < \frac{\varphi(p)}{4} \).
4. Let \( \phi \in \mathbb{D}(q)[x] \). Define \( e = p\phi \ast h + m \in \mathbb{D}(q)[x] \).
5. Suppose that \( |p\phi \ast g + f \ast m|_\infty < \frac{\varphi(q)}{4} \), where the operations to compute the polynomial are done in \( \mathbb{D}[x] \).
6. Then to decrypt, computes successively : \( a = f \ast e(\text{mod } q) \) and \( f_p \ast a(\text{mod } p) \)
7. output the plaintext \( m = f_p \ast a(\text{mod } p) \)

**Proof** Follows from section 3. \( \square \)

4.2. DTRU1: NTRU over Dual integers V.1.

Since, until know, we are not able to invert in general a polynomial \( f \) in \( \mathbb{D}(z)[x] \) where \( z \in \mathbb{D} \setminus \mathbb{Z} \) but we can do it if \( z \in \mathbb{Z} \), we are going to done a version of DTRU which we are able to implement using the arithmetic properties in [2]. For this version we consider that \( p, q \in \mathbb{Z} \) and we call the corresponding encryption algorithm **DTRU1**. This encryption scheme is the main contribution of this paper.

**Key Generation**

1. Choose \( p, q \in \mathbb{Z} \) with \( q/p \) large.
2. Choose at random a polynomial with small coefficients \( f \in \mathcal{L}_f \subset \mathbb{D}[x] \), invertible \( \text{mod } q \) and \( \text{mod } p \) and denote these inverses by \( f_q \) and \( f_p \), that is, \( f_q \ast f \equiv 1(\text{mod } q) \) and \( f_p \ast f \equiv 1(\text{mod } p) \)
3. Choose at random a polynomials with small coefficients \( g \in \mathcal{L}_g \subset \mathbb{D}[x] \).
4. Compute \( h = p.(f_q \ast g)(\text{mod } q) \in \mathbb{D}[x] \), \( h \) is the public key.

**NB:** The polynomial \( f \) and \( g \) generally have small coefficients, while \( h \) has large coefficients.

**Encryption** Suppose that Aissa wants to send a message to Oumar. Aissa do the following:

1. selecting a message \( m \) from the set of plaintexts \( \mathbb{D}(p)[x] \), where \( p \in \mathbb{Z} \): select the coefficients \( a+bc \) of \( m \) with \( a, b \in \mathcal{L}_m = \{-\frac{p-1}{2}, \ldots, -1, 0, 1, \ldots, \frac{p-1}{2} \} \)
   if \( p \) is odd and \( a, b \in \mathcal{L}_m = \{-\frac{p}{2} + 1, \ldots, -1, 0, 1, \ldots, \frac{p}{2} \} \) if \( p \) is even.
Mamadou Ghouraïssio Camara, Demba Sow and Djiby Sow

(2) chooses randomly a polynomial $\phi, \psi \in \mathcal{L}_\phi \subset \frac{D}{(x^N - 1)}$
(3) and uses the Oumar’s public key $h$ to compute

$$e = \phi * h + \psi * h^2 + m \ (mod\ q) \in \frac{D}{(x^N - 1)}$$

(4) $e$ is the encrypted message which Aissa sends to Oumar.

Decryption

Suppose that Oumar has received the message $e$ transmitted by Aissa and wants to decrypt it using his private key $f$. To do this efficiently, Oumar should have precomputed the polynomial $f_p$.

(1) In order to decrypt $e$, Oumar, first computes: $a = f^2 * e \ (mod\ q) \in \frac{D}{(x^N - 1)}$
(2) Chooses the coefficients of $a$ in the interval from $-\frac{q}{2}$ to $\frac{q}{2}$.
(3) Computing $m = f_p^2 * a \ (mod\ p) \in \frac{D}{(x^N - 1)}$

□

Example

For the example we will use much smaller parameter values:
Take $N = 7$, $p = 3$ and $q = 51$

Oumar wants to create a public/private key pair the NTRU Public Key Cryptosystem. he first randomly chooses two small polynomials $f$ and $g$ in $\frac{D}{(q)}$

1) For example

$$f = -1 + \epsilon + (1 + \epsilon)x^2 + (1 + \epsilon)x^3 + (-1 + \epsilon)x^4 + (1 + \epsilon)x^6$$

$$g = (-1 + \epsilon)x + (-1 + \epsilon)x^2 + (1 + \epsilon)x^3 + (1 + \epsilon)x^4$$

2) Oumar’s next step is to compute the inverse $f_p$ of $f$ modulo $p$ and the inverse $f_q$ of $f$ modulo $q$ using the Pseudo-Extending Euclidean Algorithm.

He find that:

$$f_p = 1 + 2\epsilon + (1 + 2\epsilon)x + (1 + 2\epsilon)x^2 + (1 + 2\epsilon)x^3 + 2\epsilon x^4 + (2 + \epsilon)x^5 + (1 + 2\epsilon)x^6$$

$$f_q = 1 + 41\epsilon + (25 + 29\epsilon)x + (13 + 23\epsilon)x^2 + (19 + 32\epsilon)x^3 + (42 + 50\epsilon)x^4 + (5 + 4\epsilon)x^5 + (49 + 20\epsilon)x^6$$

3) The final step in key creation is to compute the product

$$h = pg * f_q = 21 - 3\epsilon + (15 - 6\epsilon)x + (-6 - 3\epsilon)x^2 + (6 - 18\epsilon)x^3 + (24 - 18\epsilon)x^4 - 12x^5 + (3 + 9\epsilon)x^6$$

Oumar’s private key is the pair of polynomials $f$ and $f_p$ and his the public key is the polynomial $h$.

Encryption

Now, suppose Aissa want to send message $m = -1 + \epsilon + (1 + \epsilon)x + (-1 + \epsilon)x^2 + (1 + \epsilon)x^3 + (-1 + \epsilon)x^5$, to Oumar, using Oumar’s public key $h$

1) she chooses two random polynomials $\phi$ and $\psi$ of degree 6.

$$\phi = -1 + \epsilon + (1 + \epsilon)x + (-1 + \epsilon)x^5 + (1 + \epsilon)x^6$$

$$\psi = -1 + \epsilon + (1 + \epsilon)x + (1 + \epsilon)x^3 + (-1 + \epsilon)x^6$$
and
\[ \psi = -1 + \epsilon + (1 + \epsilon)x^3 + (-1 + \epsilon)x^4 + (1 + \epsilon)x^6 \]

2) Then her encrypted message is:
\[ e = \phi \ast h + \psi \ast h^2 + m \mod (51 + 0\epsilon) = -10 - 8\epsilon + (4 + 22\epsilon)x + (-16 - 14\epsilon)x^2 + (22 - 8\epsilon)x^3 - 12\epsilon x^4 + (-22 + 10\epsilon)x^5 + (21 + 15\epsilon)x^6 \mod (51 + 0\epsilon) \]

Decryption
Oumar has received the encrypted message \( e \) from Aissa. He uses his private key \( f \) to compute

1) \( a = f^2 \ast e = 10 - 15 + (18 + 19\epsilon)x + (-5 - 24\epsilon)x^2 + (-12 - 23\epsilon)x^3 + (-4 + 15\epsilon)x^4 + (-13 - 13\epsilon)x^5 + (8 - 21\epsilon)x^6 \mod (51 + 0\epsilon) \)

Note that when Oumar reduces the coefficients of \( f \ast e \) modulo 31 + 0\epsilon, he chooses values lying between -24 and 25. Next Oumar reduces the coefficients of \( a \) modulo 3 + 0\epsilon to get:

2) \( b = 1 + \epsilon x + x^2 + \epsilon x^3 - x^4 + (-1 - \epsilon)x^5 - x^6 \mod (3 + 0\epsilon) \)

Finally Oumar uses \( f_p \), to compute:

3) \( \overline{m} = f^*_p \ast b = -1 + \epsilon + (1 + \epsilon)x + (-1 + \epsilon)x^2 + (1 + \epsilon)x^3 + (-1 + \epsilon)x^5 \mod (3 + 0\epsilon) \)

The polynomial \( \overline{m} \) is Aissa’s message \( m \), so Oumar has successfully decrypted Aissa’s message.

4.3. Parameters of DTRU1. We need to choose parameters so that decryption is almost always possible.

So, to circumvent this problem, we decided to use small values for the coefficients of our polynomials.

1) Example for \( p = 3 \), we use values from the following set: \( \{ a + \epsilon b, a, b \in \{0, 1, -1\} \} \), and generates our polynomials according to the following rule: define the set \( \mathcal{L}(d) := \{ f | f \in \mathcal{P} \text{ with } \text{d \ coeff equal to } -1, 1, -\epsilon, \epsilon, 1 - \epsilon, 1 + \epsilon, -1 + \epsilon, -1 - \epsilon \text{ and the rest } 0 \} \). and let \( d_f, d_g, d_r \) and \( d_m \) be integers. For keys, random and messages space, we consider the following sets:

\( \mathcal{L}_f = \mathcal{L}(d_f), \mathcal{L}_g = \mathcal{L}(d_g), \mathcal{L}_r = \mathcal{L}(d_r) \) and \( \mathcal{L}_m = \mathcal{L}(d_m) \).

We must choose the size of the parameters (i.e of the sets \( \mathcal{L}_f, \mathcal{L}_g, \mathcal{L}_r, \mathcal{L}_m \)) to circumvent brute force attack and others attacks such as lattice attacks.

2) Example for \( p = 2 \), we use values from the following set: \( \{0, 1, \epsilon\} \), and generates our polynomials according to the following rule: define the set

\( \mathcal{B}(d) = \{ f \in R/f \text{ has } \text{d \ coeff equal to } 1, \epsilon, 1 + \epsilon \text{ and the rest } 0 \} \), and let \( d_f, d_g, d_r \) and \( d_m \) be integers. For keys, random and messages space, we consider the following sets: \( \mathcal{B}_f = \mathcal{B}(d_f), \mathcal{B}_g = \mathcal{B}(d_g), \mathcal{B}_r = \mathcal{B}(d_r) \) and \( \mathcal{B}_m = \mathcal{B}(d_m) \).

We must choose the size of the parameters (i.e of the sets \( \mathcal{B}_f, \mathcal{B}_g, \mathcal{B}_r, \mathcal{B}_m \)) to circumvent brute force attack and others attacks such as lattice attacks.

Brute force attacks An attacker can recover the private key by trying all possible \( f \in \mathcal{L}_f \) and testing if \( f \ast h \) (modulo q) has small entries, or by trying all \( g \in \mathcal{L}_g \) and testing if \( g \ast h^{-1} \) (modulo q) has small entries. Similarly, an
attacker can recover a message by trying all possible \( r \in \mathcal{L}_r \) and testing if \( e - pr \ast h \pmod{q} \) has small entries.

Recall that our parameters set are defined by:
\[
\mathcal{L}(d) := \{ f | f \in \mathcal{P} \text{ with d coeff equal to } -1, 1, -\epsilon, \epsilon \text{ and the rest 0} \} \text{ or}
\mathcal{B}(d) = \{ f \in R / f \text{ has d coeff equal to 1, } \epsilon \text{ and the rest 0} \}
\]
For the size we have:
\[
|\mathcal{L}(d)| = \frac{N!}{(N-4d)!} \quad |\mathcal{B}(d)| = \frac{N!}{(N-3d)!}
\]
If \( k \) is the security parameter, we must choose \( \sqrt{|\mathcal{L}(d)|} > 2^k \) or \( \sqrt{|\mathcal{B}(d)|} > 2^k \) to circumvent brute force attacks.

<table>
<thead>
<tr>
<th>Security</th>
<th>( N )</th>
<th>( p )</th>
<th>( q )</th>
<th>( d_f )</th>
<th>( d_g )</th>
<th>( d_r )</th>
</tr>
</thead>
<tbody>
<tr>
<td>medium security</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>8</td>
<td>d</td>
</tr>
<tr>
<td>high security</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>8</td>
<td>d</td>
</tr>
<tr>
<td>Very high security</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>8</td>
<td>d</td>
</tr>
</tbody>
</table>

Meet-in-the-Middle attack on NTRU encryption

If \( k \) is the security parameter, we must choose \( \frac{\sqrt{|\mathcal{L}(d)|}}{\sqrt{N}} > 2^k \) or \( \frac{\sqrt{|\mathcal{B}(d)|}}{\sqrt{N}} > 2^k \) to circumvent the most practical methods for Meet-in-the-Middle attack on NTRU public key and NTRU cipher text (see [12]).

Lattice attacks

We can quite easily generalize the lattice (see....) attack on NTRU to our system.

5. DTRU2: NTRU with non-invertible polynomial

In this generalization (see[1]) of the original NTRU cryptosystem, one selects parameters \((N, p, q)\) where \( N \) is integer and \( p \) and \( q \) are dual integers and four sets \( \mathcal{L}_f, \mathcal{L}_g, \mathcal{L}_r, \mathcal{L}_m \) of polynomials in the ring \( \mathcal{P} = \mathbb{Z}[x] / (x^N - 1) \). We denote by * the operation of multiplication in the ring \( \mathcal{P} \). The parameters \( p \) and \( q \) are distinct irreducibles dual integers with \( \varphi(q) > \varphi(p) \), and the sets \( \mathcal{L}_f, \mathcal{L}_g, \mathcal{L}_r, \mathcal{L}_m \) are chosen to satisfy the width condition \( \|pr \ast g + m \ast f\|_\infty < \frac{\varphi(q)}{4} \) for all polynomials \( f \in \mathcal{L}_f, g \in \mathcal{L}_g, r \in \mathcal{L}_r, m \in \mathcal{L}_m \) where for any polynomial
\[
F = F_0 + F_1 x^1 + ... + F_{N-1} x^{N-1},
\]
We define the width of \( F \) by \( \|F\|_\infty = \max_{0 \leq i \leq N-1} \varphi(f_i) \). Our generalization of NTRU can be described as follows.

**Key creation.** Oumar randomly selects polynomials \( f \in \mathcal{L}_f, g \in \mathcal{L}_g \) and \( G \in \mathcal{P} \) such that \( G \) has an inverse modulo \( q \) and \( f \) has an inverse modulo \( p \). Oumar first compute inverses \( G_q \) and \( f_p \) using the Pseudo-Extending Euclidean Algorithm that satisfy \( G \ast G_q \equiv 1 \pmod{q} \), \( f \ast f_p \equiv 1 \pmod{p} \).
then Oumar computes the products \( h \equiv G_q \ast q \pmod{q} \), \( H \equiv G_q \ast f \pmod{q} \). Alice publishes the pair of polynomials \((h, H)\) as her public key, retaining \((f, g, G)\) as her private key. The polynomial \( f_p \) is simply stored for later use, and the polynomial \( G_q \) may be discarded.

**Encryption.** Suppose Aissa wants to send a secret message to Oumar. Omar selects a message \( m \) from the set of plaintexts \( \mathcal{L}_m \). Next, Aissa selects a random polynomial \( r \in \mathcal{L}_r \) and uses Oumar’s public key \((h, H)\) to compute 
\[
e \equiv pr \ast h + H \ast m \pmod{q}.
\]
Aissa then transmits \( e \) to Oumar.

**Decryption.** Oumar has received \( e \) from Aissa. To decrypt the message, he first computes
\[
a \equiv G \ast e \equiv pr \ast g + f \ast m \pmod{q},
\]
choosing the coefficients of \( a \) to lie in the interval from \(-\frac{\varphi(q)}{2}\) to \(\frac{\varphi(q)}{2}\). Finally, Oumar recovers the message by computing 
\[
m \equiv f_p \ast a \pmod{p}.
\]

**Demonstration**

If the all coefficients of \( a \) is between \(-\frac{\varphi(q)}{2}\) and \(\frac{\varphi(q)}{2}\), then it checks:
\[
a \equiv G \ast e \equiv G \ast (pr \ast h + H \ast m) \pmod{q},
\]
\[
\equiv pr \ast G \ast h + G \ast H \ast m \pmod{q},
\]
\[
\equiv pr \ast G \ast (G_q \ast g) + G \ast (G_q \ast f) \ast m \pmod{q},
\]
\[
\equiv pr \ast g + f \ast m \pmod{q},
\]
Similarly, the polynomial \( b \) verifies:
\[
b \equiv f_p \ast a \equiv f_p \ast (pr \ast g) + f \ast m \pmod{p},
\]
\[
\equiv f_p \ast f \ast m \pmod{p} \equiv m \pmod{p},
\]
One easily verifies that the case \( G = f \) corresponds to the classical NTRU cryptosystem (in this case, \( H = 1 \), so the public key consists solely of the polynomial \( h \)).

6. Conclusion

Upon completion of this work we have shown that the NTRU protocol used previously on Euclidean rings can be adapted to a non-Euclidean ring.

References


[28] V. Shoup, NTL a C/C++ packages for implementing big integers and various arithmetic properties available online.


Received: November 5, 2013; Published: October 9, 2018