Spring 2019

Understanding the NTRU Cryptosystem

Benjamin Clark
bmc132@zips.uakron.edu

Understanding the NTRU Public Key Cryptosystem

Benjamin Clark

April 23, 2019

Abstract

In this paper, we will examine the NTRU Public Key Cryptosystem. The NTRU cryptosystem was created by Joseph Silverman, Jeffery Hoffstein, and Jill Pipher in 1996. This system uses truncated polynomial rings to encrypt and decrypt data. It was recently released into the public domain in 2013. This paper will describe how this cryptosystem works and give a basic understanding on how to encrypt and decrypt using this system.

Introduction

Cryptography can be traced back to at least as far as Julius Caesar and the Romans. Cryptography played a key role in the secure exchange of messages for governments and also the public. A cryptosystem involves a process of encryption, or changing a message into illegible text or numbers, and decryption, which is turning the encrypted message back into the original. Both private key cryptosystems and public key cryptosystems are available to the public, but public key cryptosystems have certain advantages over private key cryptosystems.

The NTRU cryptosystem is a public key cryptosystem that was created by Joseph Silverman, Jeffery Hoffstein, and Jill Pipher in 1996. The NTRU Public Key Cryptosystem was originally only commercially available. In 2013, it was released into the public domain for public use. The security of NTRU is based on the perceived difficulty and the prohibitive amount of time required to solve certain computational mathematical problems. These computational problems are defined by certain numerical parameters chosen

in the implementation of the cryptosystem. In this paper, we will study the NTRU cryptosystem in order to acquire a better understanding in the hopes of using this to better encrypt your messages and data. We will further discuss the security of the NTRU cryptosystem in a later section. To do this, we must take a look at truncated polynomial rings, converting messages into binary, and turning those strings of binary digits into polynomials.

1 Preliminary Background

1.1 Cryptosystems

Cryptography is the practice and study of techniques for the security of information sharing in the presence of third parties. Within cryptography are various specific cryptosystems, each of which has its own process of encryption, decryption, and key generation. With encryption, information is converted into a jumble of letters and/or numbers. Decryption is the inverse of encryption in the sense that an encrypted message is converted (or decrypted) back into its original. For each cryptosystem, there are usually one or more numerical parameters which must be specified to define the details of the encryption process. These parameters are referred to as the encryption keys. Also, there are one or more numerical parameters used to define the decryption process. These parameters are called the decryption keys. To illustrate this concept, we shall consider the Caesar Cipher. For example, the message "HELLO WORLD" can be changed (or encrypted) into "JGNNQ YQTNF" by shifting each letter over two places forward in the alphabet. In this case, the encryption key is the shift of the letters two places forward (key = 2). Following our example from before, "JGNNQ YQTNF" would be converted back into "HELLO WORLD" by shifting each letter two places backward in the alphabet. The key within this case is the inverse of our encryption key, which is shifting the letters backwards (key = -2). In this cipher, any key values ranging from 1 to 25 may be used.

The Caesar Cipher mentioned above is an example of a private key cryptosystem. In a private key cryptosystem, the key that is generated is used for both the encryption and decryption processes. As seen above, if you know the encryption key, you can figure out the decryption key very easily. Also, it is easy to figure out the encryption key if the decryption key is known. Thus, it is obvious that both the encryption and decryption keys are meant to remain secret to the sender and receiver. This is why it is called a private key cryptosystem.

In contrast, there is the notion of a public key cryptosystem. In a public key cryptosystem, an encryption key and decryption key are generated. The encryption key is published to the world, and this is known as the public key. The decryption key is to remain secret to the owner of the keys. This is the private key. Note that the public and private keys are different. As mentioned earlier, NTRU is an example of a public key cryptosystem.

1.2 Modular Arithmetic

Let m be a positive integer and let a be any integer. The expression

$$a\,(modulo\,m)$$

denotes the unique nonnegative integer r such that $0 \leq r < m$ while $a - r$ is an integer multiple of m. Also, the statement

$$a \equiv b\,(modulo\,m)$$

asserts that a and b both have the same remainder when divided by m. Thus, the difference of $a-b$ is a multiple of m. The term given to m is the modulus.

Modular arithmetic can also have the operations of addition, subtraction, and multiplication performed on them. For any integers a and b, we have

$$\begin{aligned} a\,(modulo\,m) + b\,(modulo\,m) &\equiv a + b\,(modulo\,m) \\ a\,(modulo\,m) - b\,(modulo\,m) &\equiv a - b\,(modulo\,m) \\ a\,(modulo\,m) \times b\,(modulo\,m) &\equiv ab\,(modulo\,m) \end{aligned}$$

We say that two integers a and m are relatively prime if their only common positive integer factor is 1. If a and m are relatively prime, there exists an integer b (also known as an inverse of a modulo m) such that

$$ab \equiv 1\,(modulo\,m).$$

In the NTRU cryptosystem, modular arithmetic is performed on polynomials. Thus, these properties can be extended to polynomials. Let $\mathbf{a}$ and $\mathbf{b}$ be polynomials of the form:

$$\begin{aligned} \mathbf{a} &= a_0 + a_1X + a_2X^2 + ... + a_{N-1}X^{N-1} + a_NX^N \\ \mathbf{b} &= b_0 + b_1X + b_2X^2 + ... + b_{N-1}X^{N-1} + b_NX^N, \end{aligned}$$

where all coefficients are integers. We write **ab** to denote the the product of the polynomials **a** and **b**. The expression

$$\mathbf{ab}(modulo\, m)$$

refers to the polynomial obtained when each coefficient c of **ab** is replaced by $c\,(modulo\, m)$. For example, take

$$\begin{aligned}\mathbf{a} &= 2 + 5X^3\\ \mathbf{b} &= X + X^2 + 6X^3\\ m &= 2.\end{aligned}$$

First, we multiply the polynomials:

$$\mathbf{ab} = 2X + 2X^2 + 12X^3 + 5X^4 + 5X^5 + 30X^6$$

Then we replace each coefficient by its remainder upon division by 2. Thus,

$$\mathbf{ab}\,(modulo\, 2) = X^4 + X^5.$$

1.3 Truncated Polynomial Rings

For an arbitrary positive integer k, let

$$\mathbf{Z}_k = \{0, 1, ..., k-1\}.$$

denote the ring of integers modulo k. Fix a positive integer N. Let **T** be the set of polynomials of degree at most $N-1$ with coefficients in $\mathbf{Z}_k$. Let **a** and **b** be two elements from **T**. These have the form:

$$\begin{aligned}\mathbf{a} &= a_0 + a_1X + a_2X^2 + ... + a_{N-2}X^{N-2} + a_{N-1}X^{N-1}\\ \mathbf{b} &= b_0 + b_1X + b_2X^2 + ... + b_{N-2}X^{N-2} + b_{N-1}X^{N-1}\end{aligned}$$

where $a_0, ..., a_{N-1}\, and\, b_0, ..., b_{N-1} \in \mathbf{Z}_k$.

Addition within this set is performed in the usual way by adding the coefficients of the elements. For example:

$$\mathbf{a} + \mathbf{b} = (a_0 + b_0)X + (a_1 + b_1)X^1 + ... + (a_{N-1} + b_{N-1})X^{N-1}.$$

It is obvious that $\mathbf{a} + \mathbf{b} \in \mathbf{T}$.

For a given pair of polynomials **a** and **b** in **T**, the usual way of performing multiplication may cause the resulting product **ab** to have a degree greater than $N-1$, and thus not belong to **T** . We wish to define the product **ab** in such a way that **ab** is in **T**. It is obvious that the product of two polynomials of degree N will create a polynomial with a degree greater than $N-1$. For each integer m greater than $N-1$, we replace X^m by X^r, where r is the least nonnegative residue of

$$m \, modulo \, N.$$

The coefficients are added in the usual way. It is obvious that $\mathbf{ab} \in \mathbf{T}$.

For example, let

$$\begin{aligned} N &= 5 \\ \mathbf{a} &= 2 + X + X^3 \\ \mathbf{b} &= 8X + 5X^2 + 2X^4. \end{aligned}$$

Thus,

$$\begin{aligned} \mathbf{ab} &= (2 + X + X^3)(8X + 5X^2 + 2X^4) \\ &= 16X + 10X^2 + 4X^4 + 8X^2 + 5X^3 + 2X^5 + 8X^4 + 5X^5 + 2X^7 \\ &= 16X + 18X^2 + 5X^3 + 12X^4 + 7X^5 + 2X^7 \notin \mathbf{T}. \end{aligned}$$

By following the rules before, we get

$$\begin{aligned} &= 16X + 18X^2 + 5X^3 + 12X^4 + 7X^5 + 2X^7 \\ &= (7+0)X^0 + 16X + (18+2)X^2 + 5X^3 + 12X^4 \\ &= 7 + 16X + 20X^2 + 5X^3 + 12X^4 \in \mathbf{T}. \end{aligned}$$

We define these polynomials as truncated polynomials. It is easy to see that **T** has two binary operations of addition and multiplication and these operations satisfy the properties of associativity and distributivity and thus **T** is a ring.

1.4 Inverses of Truncated Polynomials

For the purpose of this cryptosystem, we must find the inverse of a polynomial relative to a specified modulus. Let $\mathbf{p} \in \mathbf{T}$ and q be an integer. It is sometimes possible to find a polynomial $\mathbf{P} \in \mathbf{T}$ such that

$$\mathbf{pP} \equiv 1 \, (modulo \, q).$$

Note that a chosen polynomial $\mathbf{p}$ might not have an inverse. For example, let

$$\begin{aligned}\mathbf{p} &= X + X^2\\ \mathbf{P} &= 2 + X\\ q &= 2\\ N &= 3\end{aligned}$$

Thus,

$$\begin{aligned}\mathbf{pP}(modulo\,2) &= (X + X^2)(2 + X)\\ &= 2X + X^2 + X^2 + X^3\\ &= 2X + 2X^2 + X^3\\ &= 1 + 2X + 2X^2\\ &= 1\end{aligned}$$

1.5 Letter to Binary Conversion

With computers, all messages, words, pictures, etc. are converted into a string of ones and zeros using ASCII, or the American Standard Code for Information Interchange. For the NTRU cryptosystem, the premise is the same. Therefore, a basic knowledge of this conversion is necessary. This conversion is relatively simple and always has the same output, so there are tables that can be used. One will be provided in the appendix for our use.

2 NTRU Public Key Cryptosystem

2.1 Parameters

Several different versions of the NTRU cryptosystem have been created, some of which are more complicated than others for the sake of added security. In this paper, we examine perhaps the simplest version of NTRU, in which only three parameters are used. Note that some of the more complex versions use more parameters. This cryptosystem utilizes the set $\mathbf{T}$ defined in section 1.3.

The three parameters we will be using are

- N the polynomials in the ring defined above.
- q a chosen large modulus.
- p a chosen small modulus.

2.2 Generating a Key

Suppose Bob wants to send a message to Alice using the NTRU cryptosystem. Before he can do this, Alice must first create her encryption key and her decryption key. Alice randomly chooses two polynomials $\mathbf{f}$ and $\mathbf{g}$ from our ring $\mathbf{T}$. These polynomials both need to be "small" relative to a random polynomial modulo q. This means that the coefficients do not have coefficients greater than q.

Alice's next step is to compute the inverse of $\mathbf{f}$ with respect to modulo q and modulo p respectively. If $\mathbf{f}$ has an inverse with respect to these moduli, they will satisfy the properties

$$\mathbf{f}\mathbf{f}_q = 1\,(modulo\,q) \tag{1}$$

and

$$\mathbf{f}\mathbf{f}_p = 1\,(modulo\,p), \tag{2}$$

where $\mathbf{f}_q$ is the inverse of $\mathbf{f}$ modulo q and $\mathbf{f}_p$ is the inverse of $\mathbf{f}$ modulo p. Note that if either $\mathbf{f}_q$ or $\mathbf{f}_p$ does not exist, Alice must choose another $\mathbf{f}$ repeatedly until both of these inverses exist.

After Alice chooses an $\mathbf{f}$ for which both inverses exist, she must compute the polynomial $\mathbf{h}$ defined by:

$$\mathbf{h} = p\mathbf{f}_q\mathbf{g}\,(modulo\,q)$$

Now Alice's private key is the pair of polynomials $\mathbf{f}_p$ and $\mathbf{f}$, which need to remain secret. Her public key is the polynomial $\mathbf{h}$, which she can then publish to the world.

2.3 Encryption

Now that Alice has created her encryption and decryption key, she can publish her public key $\mathbf{h}$. Bob can now send a message to Alice. Bob must first convert his message into the form of a polynomial $\mathbf{m}$ whose coefficients are chosen modulo p so that $\mathbf{m}$ becomes a small polynomial modulo q. Bob also must choose another random small polynomial $\mathbf{r}$ which will then be used to obscure the message. This will make it harder to be decrypted by a person who is not the intended recipient.

Bob uses his polynomials $\mathbf{m}$ and $\mathbf{r}$ and Alice's public key $\mathbf{h}$ to compute the polynomial $\mathbf{e}$ defined by

$$\mathbf{e} = \mathbf{rh} + \mathbf{m}\,(modulo\, q),$$

and this is now Bob's encrypted message. Bob can now send the polynomial $\mathbf{e}$ to Alice.

2.4 Decryption

Now that Alice has received Bob's message, she wants to decrypt it to see what the message holds. Alice now uses her polynomial $\mathbf{f}$ to compute the polynomial $\mathbf{a}$ defined by

$$\mathbf{a} = \mathbf{fe}\,(modulo\, q).$$

Next, she computes the polynomial $\mathbf{b}$ defined by

$$\mathbf{b} = \mathbf{a}\,(modulo\, p).$$

Lastly, Alice computes the polynomial $\mathbf{c}$ defined by

$$\mathbf{c} = \mathbf{f}_p\mathbf{b}\,(modulo\, p).$$

This polynomial $\mathbf{c}$ will be Bob's original message $\mathbf{m}$. We shall now explain why.

Recall our ring, $\mathbf{T}$. This, along with the existence of an inverse, allows Alice to figure out Bob's message. When $\mathbf{a}$ is computed, Alice is actually computing

$$\begin{aligned}
\mathbf{a} &= \mathbf{fe}(modulo\, q) \\
&= \mathbf{f}(\mathbf{rh} + \mathbf{m})\,(modulo\, q) \\
&= \mathbf{f}(\mathbf{r}p\mathbf{f}_q\mathbf{g} + \mathbf{m})\,(modulo\, q) \\
&= \mathbf{ff}_q p\mathbf{rg} + \mathbf{fm}\,(modulo\, q) \\
&= p\mathbf{rg} + \mathbf{fm}\,(modulo\, q)
\end{aligned}$$

Remember that all the coefficients for the polynomials are small in comparison to q. Also, p is small relative to q. Therefore, when reducing by modulo q, no reduction is needed because the coefficients will remain the same. Since

$$\mathbf{a} = p\mathbf{rg} + \mathbf{fm},$$

then

$$\begin{aligned}\mathbf{b} &= \mathbf{a}\,(modulo\,p)\\ &= p\mathbf{rg} + \mathbf{fm}\,(modulo\,p)\\ &= \mathbf{fm}\,(modulo\,p).\end{aligned}$$

Now, remember that

$$\mathbf{c} = \mathbf{f}_p\mathbf{b}\,(modulo\,p),$$

and since $\mathbf{b} = \mathbf{fm}\,(modulo\,p)$, we have

$$\begin{aligned}\mathbf{c} &= \mathbf{f}_p\mathbf{b}\,(modulo\,p)\\ &= \mathbf{f}_p\mathbf{fm}\,(modulo\,p)\\ &= \mathbf{m}\,(modulo\,p)\\ &= \mathbf{m}.\end{aligned}$$

Thus, Alice has the original message m.

2.5 An Example of Using NTRU

To begin using the NTRU cryptosystem, we must first choose integers N, p, and q. Let $N = 11$, $q = 32$, and $p = 3$. Now we must choose a polynomial $\mathbf{f}$ so that $\mathbf{f}_p$ and $\mathbf{f}_q$ exist and a polynomial $\mathbf{g}$. Let

$$\mathbf{f} = -1 + X + X^2 - X^4 + X^6 + X^9 - X^{10}$$

and

$$\mathbf{g} = -1 + X + X^2 + X^3 - X^8 - X^{10}.$$

The corresponding inverses of $\mathbf{f}$ are

$$\mathbf{f}_p = 1 + 2X + 2X^3 + 2X^4 + X^5 + 2X^7 + X^8 + 2X^9$$

and

$$\mathbf{f}_q = 5+9X+6X^2+16X^3+4X^4+15X^5+16X^6+22X^7+20X^8+18X^9+30X^{10}.$$

Recall that $\mathbf{h} = p\mathbf{f}_q\mathbf{g}\,(modulo\,q)$. Using this, we compute

$$\begin{aligned}\mathbf{h} &= p\mathbf{f}_q\mathbf{g}\,(modulo\,q)\\ &= 18 + 6X + 21X^2 + 16X^3 + 2X^4 + 21X^5 + X^6 + 17X^7 + 30X^8 + 3X^9 + 25X^{10}.\end{aligned}$$

Now that we have $\mathbf{h}$, we can publish it to the world to allow messages to be sent back to us.

Now let us send ourselves a message. Take the message "Hi". To convert this into a polynomial, we must first convert the letters into their respective binary representations. This can be done using the table in the appendix. Thus,

$$H = 01001000$$
$$i = 01101100$$

and to convert them into polynomials, we just add an X to the respective powers as follows

$$H = X^3 + X^6$$
$$i = 1 + X^3 + X^5 + X^6$$

Let these be the polynomials $\mathbf{m}_1$ and $\mathbf{m}_2$ respectively.

Before we can use the public key $\mathbf{h}$ to encrypt the message, we must choose a random polynomial $\mathbf{r}$. Let

$$\mathbf{r} = 1 + X + X^2 + X^7.$$

Now, recall our encrypted message is defined by

$$\mathbf{e} = \mathbf{rh} + \mathbf{m}_1 \,(modulo\, q).$$

Thus, using $\mathbf{m}_1$, we can compute $\mathbf{e}$:

$$\begin{aligned}\mathbf{e} &= \mathbf{rh} + \mathbf{m}_1 \,(modulo\, q) \\ &= 14 + 6X + 14X^2 + 29X^3 + 5X^4 + 18X^5 + 18X^6 + 7X^7 + 10X^8 + 7X^9 + 22X^{10}.\end{aligned}$$

To decrypt $\mathbf{m}_1$, we must compute $\mathbf{a}$. Recall $\mathbf{a}$ was defined by

$$\mathbf{a} = \mathbf{fe} \,(modulo\, q),$$

so using $\mathbf{f}$ and $\mathbf{e}$ we can compute $\mathbf{a}$:

$$\begin{aligned}\mathbf{a} &= \mathbf{fe} \,(modulo\, q) \\ &= 6 + 5X + 2X^2 + 8X^3 + 5X^4 + 3X^5 + 29X^6 + 3X^7 + X^8 + X^9 + 28X^{10}.\end{aligned}$$

Recall that $\mathbf{b} = \mathbf{a} \,(modulo\, p)$, so we can compute $\mathbf{b}_1$:

$$\begin{aligned}\mathbf{b} &= \mathbf{a} \,(modulo\, p) \\ &= 2X + 2X^2 + 2X^3 + 2X^4 + 2X^6 + X^8 + X^9 + X^{10}\end{aligned}$$

Recall that $\mathbf{c} = \mathbf{f}_p\mathbf{b}\,(modulo\,p)$, so now we calculate c:

$$\begin{aligned}\mathbf{c}_1 &= \mathbf{f}_p\mathbf{b}(modulo\,p)\\ &= X^3 + X^6 = \mathbf{m}_1.\end{aligned}$$

We now repeat this process for $\mathbf{m}_2$. We will use the same polynomial $\mathbf{r}$ as before. Thus, the encryption would be

$$\begin{aligned}\mathbf{e} &= \mathbf{rh} + \mathbf{m}_2\,(modulo\,q)\\ &= 17 + 6X + 14X^2 + 29X^3 + 5X^4 + 11X^5 + 18X^6 + 25X^7 + 22X^8 + 7X^9 + 10X^{10}.\end{aligned}$$

To decrypt $\mathbf{m}_2$, we must compute $\mathbf{a}$ once again.

$$\begin{aligned}\mathbf{a} &= \mathbf{fe}\,(modulo\,q)\\ &= 6 + 3X^2 + 18X^3 + 3X^4 + 5X^5 + 14X^6 + 25X^7 + X^8 + X^9 + 8X^{10}.\end{aligned}$$

Thus,

$$\begin{aligned}\mathbf{b} &= \mathbf{a}\,(modulo\,p)\\ &= 2X^5 + X^6 + X^7 + X^8 + X^9 + X^{10}.\end{aligned}$$

Finally,

$$\begin{aligned}\mathbf{c} &= \mathbf{f}_p\mathbf{b}\,(modulo\,p)\\ &= 1 + X^3 + X^5 + X^6 = \mathbf{m}_2.\end{aligned}$$

We now have both portions of the message.

2.6 Security of NTRU

The security of NTRU is based on the perceived difficulty and the prohibitive amount of time required to solve certain computational mathematical problems. These computational problems are defined by the numerical parameters N, q, and p that are chosen in the implementation of the cryptosystem. The level of security can be increased by imposing certain conditions on these parameters.

The only feasible attack on this cryptosystem is known as a lattice-based attack. This involves setting up a system of equations in an attempt to solve for any chosen polynomial. This system of equations requires a matrix of size N by N. There are a few lattice-based attacks currently known, and the most complex attack can break the NTRU in about forty minutes only if N is less than 100.

Although lattice-based attacks can prove to be effective for N less than 100, certain conditions can be put in place to increase the security of NTRU. One such condition is to make N very large. If N is 200, that would supply moderate security. If N is 500, it would supply high security. This is because although a computer can solve large matrices, it becomes infeasible to break this cryptosystem if it takes weeks or even years. For example, the estimated breaking time for $N = 500$ would be about 8.4 years. Also, if p and q are chosen to be large and relatively prime, then solving this system becomes even harder to solve.

3 Appendix

ASCII conversions of letters to binary:

A	1000001		a	1100001
B	1000010		b	1100010
C	1000011		c	1100011
D	1000100		d	1100100
E	1000101		e	1100101
F	1000110		f	1100110
G	1000111		g	1100111
H	1001000		h	1101000
I	1001001		i	1101001
J	1001010		j	1101010
K	1001011		k	1101011
L	1001100		l	1101100
M	1001101		m	1101101
N	1001110		n	1101110
O	1001111		o	1101111
P	1010000		p	1110000
Q	1010001		q	1110001
R	1010010		r	1110010
S	1010011		s	1110011
T	1010100		t	1110100
U	1010101		u	1110101
V	1010110		v	1110110
W	1010111		w	1110111
X	1011000		x	1111000
Y	1011001		y	1111001
Z	1011010		z	1111010

4 Bibliography

Gcn. (n.d.). NTRU crypto software available to open-source community. Retrieved March 20, 2019, from https://gcn.com/articles/2013/12/05/ntru-crypto-software.aspx

Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman. NTRU: A Ring Based Public Key Cryptosystem. In Algorithmic Number Theory (ANTS III), Portland, OR, June 1998, J.P. Buhler (ed.), Lecture Notes in Computer Science 1423, Springer-Verlag, Berlin, 1998.

May, Alexander. Cryptanalysis of NTRU. citeseerx.ist.psu.edu.

NTRU Post Quantum Cryptography. (n.d.). Retrieved March 20, 2019, from https://www.onboardsecurity.com/products/ntru-crypto

Singh, S. (1999). The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography. New York: Anchor Books.

Stallings, W. (2017). Cryptography and Network security: Principles and Practice. Boston: Pearson Education.