The NTRU Public-key Cryptosystem

Chris Powell

1 Introduction

No known classical alogirithm can efficiently solve the *integer factorization problem* or the *discrete log problem*. For this reason, these two problems are central to the construction of all currently implemented public-key cryptosystems. But a quantum algorithm, known as *Shor's algorithm*, can efficiently solve both of these problems [Sho97]. Thus quantum computers pose a serious threat to IT security as they are capable of trivially breaking the most widely adopted asymmetric ciphers (e.g., RSA and elliptic curve cryptography). To prepare for the advent of a general purpose quantum computer, standardization groups such as the National Institute of Standards and Technology (NIST) have called for adoption of cryptosystems which are resistant to attacks by quantum computers [MCJ+16].

The NTRU cryptosystem, first introduced in [HPS98] by Hoffstein, Pipher, and Silverman, is a lattice-based cryptosytem that is resistant to attacks by both classical and quantum computers. Its security is based on the conjectured intractability of a problem in lattice reduction known as the *shortest vector problem* [PHP08].[1] NTRU consists of two cryptographic primitives: the NTRUEncrypt[2] algorithm, which is used for encryption, and NTRUSign, which is for digital signatures. This paper focuses soley on NTRUEncrypt. We give a brief description of some of its underlying mathematical constructions and show that for suitably chosen parameters, decryption of the ciphertext always matches the original plaintext.

2 Background

Definition 1 (Encryption)**.** Let $\mathcal{M}$ be the set of plaintexts, $\mathcal{C}$ the set of ciphertexts, and $\mathcal{K}_1, \mathcal{K}_2$ the keyspaces. Fix $(k_{\mathsf{pub}}, k_{\mathsf{priv}}) \in \mathcal{K}_1 \times \mathcal{K}_2$. Let $E_{k_{\mathsf{pub}}} : \mathcal{M} \to \mathcal{C}$ and $D_{k_{\mathsf{priv}}} : \mathcal{C} \to \mathcal{M}$. Then $E_{k_{\mathsf{pub}}}$ is an *encryption* if for each $m \in \mathcal{M}$,

(i) $D_{k_{\mathsf{priv}}}\left(E_{k_{\mathsf{pub}}}(m)\right) = m$,

[1] The connection with lattices is beyond the scope of this paper. For information on this topic, see discussion in [PHP08, §6.11].

[2] NTRUEncrypt is currently available under an open-source license; specifications for its implemention in C are available at `https://github.com/NTRUOpenSourceProject/NTRUEncrypt`.

(ii) $E_{k_{\mathsf{pub}}}(m)$ can be computed efficiently given k_{pub}, and

(iii) $D_{k_{\mathsf{priv}}}(c)$ can be computed efficiently given k_{priv}.

Definition 2 (Ring)**.** Let R be a set equipped with binary operations $+, \cdot : R \times R \to R$. Then R is a *ring* if

(i) $(R, +)$ forms an abelian group,

(ii) $(R, \cdot)$ forms a monoid, and

(iii) The distributive law holds, i.e., for all $a, b, c \in R$, $c \cdot (a+b) = (c \cdot a) + (c \cdot b)$.

Furthermore, if R is commutative with respect to $\cdot$, then R is called a *commutative ring.*

Remark. A monoid is like a group without the requirement that every element have an inverse.

Definition 3 (Ring homomorphism)**.** Let $\varphi : R_1 \to R_2$, where R_1 and R_2 are rings. Then φ is a *ring homomorphism* if for all $a, b \in R_1$,

(i) $\varphi(a + b) = \varphi(a) + \varphi(b)$,

(ii) $\varphi(a \cdot b) = \varphi(a) \cdot \varphi(b)$, and

(iii) $\varphi(1_{R_1}) = 1_{R_2}$.

Definition 4 (N^{th} root of unity)**.** Fix $N \in \mathbb{Z}_{>0}$. Let $\zeta \in \mathbb{C}$. Then ζ is an *N^{th} root of unity* if $\zeta = e^{\frac{2k\pi i}{N}}$ for some $k \in \{0, \ldots, N-1\}$.

Proposition 1. If ζ is an N^{th} root of unity, then $\zeta^N = 1$.

Proof. Since $2\pi \in \mathbb{R}$, Euler's identity implies

$$e^{2\pi i} = \cos(2\pi) + i \sin(2\pi) = 1.$$

Thus

$$\zeta^N = \left(e^{\frac{2k\pi i}{N}}\right)^N = e^{2k\pi i} = \left(e^{2\pi i}\right)^k = 1^k = 1.$$

■

Definition 5 (Primitive N^{th} root of unity)**.** An N^{th} root of unity ζ is *primitive* if $\zeta^k \neq 1$ for all $k \in \{1, \ldots, N-1\}$.

Example. The 4^{th} roots of unity are $\{\pm 1, \pm i\}$. But only $\pm i$ are primitive as $1^1 = 1$ and $(-1)^2 = 1$.

Proposition 2 (Cyclotomic polynomial rings)**.** Let ζ be a primitive N^{th} root of unity. Then

$$\mathbb{Z}[\zeta] = \left\{ \sum_{i=0}^{N-1} \alpha_i \zeta^i \mid \alpha_i \in \mathbb{Z} \right\} \quad \text{and} \quad \mathbb{Z}_q[\zeta] = \left\{ \sum_{i=0}^{N-1} \alpha_i \zeta^i \mid \alpha_i \in \mathbb{Z}_q \right\}$$

are commutative rings under polynomial addition and multiplication.[3]

Proposition 3. The reduction map $\mathbb{Z}[\zeta] \to \mathbb{Z}_q[\zeta]$ defined by

$$\sum_{i=0}^{N-1} \alpha_i \zeta^i \mapsto \sum_{i=0}^{N-1} [\alpha]_q \zeta^i.$$

is a ring homomorphism, i.e., for all $a, b \in \mathbb{Z}[\zeta]$,

$$[a+b]_q = [a]_q + [b]_q \quad \text{and} \quad [a \cdot b]_q = [a]_q \cdot [b]_q.$$

Remark. We use $+$ interchangably to denote addition in both $\mathbb{Z}[\zeta]$ and $\mathbb{Z}_q[\zeta]$; likewise for multiplication $\cdot$.

Proposition 4 (Proposition 6.45 in [PHP08])**.** Let q be prime. Assume

$$\gcd\left([a]_q , \zeta^N - 1 \right) = [1]_q .$$

Then we can find a polynomial $u \in \mathbb{Z}[\zeta]$ such that

$$[a]_q \cdot [u]_q = [1]_q .$$

Remark. We can compute $[u]_q = [a]_q^{-1}$ via the extended Euclidean algorithm in $\mathbb{Z}_q[\zeta]$. For the full details on this procedure, see [PHP08, page 391].

Definition 6 (Centered Lift, page 390 in [PHP08])**.** The *centered lift of a modulo q to $\mathbb{Z}[\zeta]$* is the map $\mathrm{lift}_q : \mathbb{Z}_q[\zeta] \to \mathbb{Z}[\zeta]$ which sends $[a]_q \in \mathbb{Z}_q[\zeta]$ to the unique polynomial $a' \in \mathbb{Z}[\zeta]$ satisfying $[a']_q = [a]_q$, where each coefficient α'_i of a' lies in the interval $\left(\frac{-q}{2}, \frac{q}{2} \right]$.

Definition 7 (Ternary Polynomials, page 392 in [PHP08])**.** The set of *ternary polynomials $\mathcal{T}(m,n) \subseteq \mathbb{Z}[\zeta]$* is defined to be the set of all $a(\zeta) \in \mathbb{Z}[\zeta]$ for which

$$a(\zeta) \text{ has} \begin{cases} m \text{ coefficients } \alpha_i = 1 \\ n \text{ coefficients } \alpha_i = -1 \\ N - (m+n) \text{ coefficients } \alpha_i = 0 \end{cases} .$$

[3]In the more general context of algebraic number theory, $\mathbb{Z}[\zeta]$ is known as the *ring of integers* of the cyclotomic number field $\mathbb{Q}(\zeta)$. The ring of integers is a generalization of $\mathbb{Z} \subseteq \mathbb{Q}$.

3 Description of NTRUEncrypt

Suppose Rachael would like to transmit a confidential message to Deckard, but their mutual adversary, Tyrell, has a quantum computer. To communicate securely, they agree to use NTRUEncrypt. Since Deckard is the intended recipient of Rachael's message, protocol requires that he first generate a public-private keypair. He proceeds as follows.

3.1 Key Generation

Following the procedure described in [HPS98], Deckard selects positive integers N, p, q, and d such that

$$N \text{ is an odd prime}, \quad \gcd(p,q) = 1, \quad \text{and} \quad q > p. \tag{$\circledast$}$$

Then he finds a ternary polynomial $f \in \mathcal{T}(d+1, d)$ such that f is invertible in both $\mathbb{Z}_p[\zeta]$ and $\mathbb{Z}_q[\zeta]$. He computes $[f]_p^{-1}$ and $[f]_q^{-1}$ and randomly selects a ternary polynomial $g \in \mathcal{T}(d,d)$. Deckard then hides $k_{\mathsf{priv}} = \left(f, [f]_p^{-1}\right) \in \mathbb{Z}[\zeta] \times \mathbb{Z}_p[\zeta]$ and publishes $k_{\mathsf{pub}} = [f]_q^{-1} \cdot [g]_q \in \mathbb{Z}_q[\zeta]$, along with the parameters (N, p, q, d).

Example 1. Consider NTRU public parameters $(N, p, q, d) = (5, 3, 41, 2)$. Let $f \in \mathcal{T}(3,2)$ and $g \in \mathcal{T}(2,2)$ be the polynomials

$$f(\zeta) = \zeta^4 - \zeta^3 + \zeta^2 - \zeta + 1 \quad \text{and} \quad g(\zeta) = \zeta^4 + \zeta^3 - \zeta^2 - \zeta.$$

Applying the extended Euclidean algorithm in $\mathbb{Z}_q[\zeta]$ and $\mathbb{Z}_p[\zeta]$, we find

$$[f(\zeta)]_3^{-1} = [2\zeta + 2]_3 \quad \text{and} \quad [f(\zeta)]_{41}^{-1} = [21\zeta + 21]_{41}.$$

Next we compute the product

$$\begin{aligned} [f(\zeta)]_{41}^{-1} \cdot \big[g(\zeta)\big]_{41} &= [21\zeta + 21]_{41} \cdot [\zeta^4 + \zeta^3 + 40\zeta^2 + 40\zeta]_{41} \\ &= \left[\zeta^4 + 40\zeta^2 + 20\zeta + 21\right]_{41}. \end{aligned}$$

Thus $k_{\mathsf{pub}} = \left[\zeta^4 + 40\zeta^2 + 20\zeta + 21\right]_{41}$ and $k_{\mathsf{priv}} = \left(\zeta^4 - \zeta^3 + \zeta^2 - \zeta + 1, [2\zeta + 1]_3\right)$.

3.2 Encryption

As observed in [PHP08, page 393], the NTRU plaintext space is

$$\mathcal{M} = \left\{ \sum_{i=0}^{N-1} \alpha_i \zeta^i \in \mathbb{Z}[\zeta] \mid |\alpha_i| \leq \frac{p}{2} \right\} \subseteq \mathbb{Z}[\zeta].$$

So upon obtaining Deckard's public key k_{pub} and choice of parameters (N, p, q, d), Rachael encodes her message as a polynomial $m \in \mathcal{M}$ and generates an ephemeral key $r \in \mathcal{T}(d, d)$.[4] She then applies the NTRU encryption $E_{k_{\mathsf{pub}}} : M \to \mathbb{Z}_q[\zeta]$ defined in [HPS98] by

$$E_{k_{\mathsf{pub}}}(m) = [p \cdot k_{\mathsf{pub}} \cdot r + m]_q.$$

Finally, Rachael transmits the ciphertext $c = E_{k_{\mathsf{pub}}}(m)$ to Deckard.

Example 2. Let (N, p, q, d), f, and g be as in Example 1. Let $m \in M$ be the polynomial

$$m(x) = \zeta^4 - \zeta^3 + \zeta + 1.$$

Let the ephemeral key $r \in \mathcal{T}(2, 2)$ be the polynomial

$$r(x) = -\zeta^3 + \zeta^2 + \zeta - 1.$$

Then the NTRU ciphertext $c \in \mathbb{Z}_{41}[\zeta]$ is the polynomial

$$\begin{aligned} E_{k_{\mathsf{pub}}}(m) &= [3]_{41} \cdot \left[\zeta^4 + 40\zeta^2 + 20\zeta + 21\right]_{41} \cdot \left[-\zeta^3 + \zeta^2 + \zeta - 1\right]_{41} + \left[\zeta^4 - \zeta^3 + \zeta + 1\right]_{41} \\ &= \left[17\zeta^4 + 34\zeta^3 + 7\zeta + 26\right]_q. \end{aligned}$$

3.3 Decryption

Upon receiving Rachael's ciphertext $c \in \mathbb{Z}_q[\zeta]$, Deckard applies decryption $D_{k_{\mathsf{priv}}} : \mathbb{Z}_q[\zeta] \to M$ given in [HPS98] by

$$D_{k_{\mathsf{priv}}}(c) = \operatorname{lift}_p\left([f]_p^{-1} \cdot \left[\operatorname{lift}_q\left([f]_q \cdot [c]_q\right)\right]_p\right).$$

Example 3. Let (N, p, q, d), f, g, m, and c be as above. We first compute the product

$$\begin{aligned} [f]_{41} * [c]_{41} &= \left[\zeta^4 - \zeta^3 + \zeta^2 - \zeta + 1\right]_{41} \cdot \left[17\zeta^4 + 34\zeta^3 + 7\zeta + 26\right]_{41} \\ &= \left[2\zeta^4 + 32\zeta^3 + 36\zeta^2 + 5\zeta + 9\right]_{41}. \end{aligned}$$

By center lifting modulo 41, we obtain the polynomial

$$2\zeta^4 + 9\zeta^3 - 5\zeta^2 + 5\zeta + 9 \in \mathbb{Z}[\zeta].$$

Next, we compute

$$[2\zeta + 2]_3 \cdot \left[2\zeta^4 + 9\zeta^3 - 5\zeta^2 + 5\zeta + 9\right]_3 = \left[\zeta^4 + 2\zeta^3 + \zeta + 1\right]_3.$$

Finally, by center lifting modulo 3 to $\mathbb{Z}[\zeta]$, we recover the plaintext

$$m(\zeta) = \zeta^4 - \zeta^3 + \zeta + 1.$$

[4] An *ephemeral key* is cryptographic key that is used once and then discarded; it is not stored.

4 Main Results

We now show that for suitable parameters (N, p, q, d), decryption of the ciphertext always matches the original plaintext. The argument for the following results closely follows the treatment given in [PHP08].

Lemma 4.1. Let N, p, q, and d be positive integers satsifying $\circledast$. Let $f \in \mathcal{T}(d+1, d)$, $g, r \in \mathcal{T}(d, d)$, and $m \in \mathcal{M}$. Write $p \cdot g \cdot r + f \cdot m = \sum \alpha_i \zeta^i$. Then

$$|\alpha_i| < \left(3d + \frac{1}{2}\right) \cdot p$$

for all i.

Proof. Observe that since $g, r \in \mathcal{T}(d, d)$, each has exactly d coefficients equal to 1 and d coeffcients equal to -1. It follows that each coefficient of the product $g \cdot r$ has magnitude at most $2d$. Similarly, $f \in \mathcal{T}(d+1, d)$ has exactly $d+1$ coefficients equal to 1 and d coefficients equal to -1, and each coefficient of $m \in \mathcal{M}$ has magnitude at most $\frac{1}{2}$, by definition of $\mathcal{M}$. Thus each coefficient in the product $f \cdot m$ has magnitude at most $(2d+1) \cdot \frac{1}{2}$. Together this implies

$$|\alpha_i| \leq p \cdot \left((2d) + (2d+1) \cdot \frac{1}{2}\right) = \left(3d + \frac{1}{2}\right) \cdot p$$

for all i. ∎

Lemma 4.2. If (N, p, q, d) satisfies $q > (6d+1) \cdot p$, then

$$\operatorname{lift}_q \left([p \cdot g \cdot r + f \cdot m]_q \right) = p \cdot g \cdot r + f \cdot m.$$

Proof. Write $p \cdot g \cdot r + f \cdot m = \sum \alpha_i \zeta^i$. Since

$$q > (6d+1) \cdot p = 2 \cdot (3d + \tfrac{1}{2}) \cdot p,$$

the previous lemma implies that $|\alpha_i| < \frac{q}{2}$ for all i. In other words, each coefficient of $p \cdot g \cdot r + f \cdot m$ is contained in the interval $(-\frac{q}{2}, \frac{q}{2})$. This proves the lemma. ∎

Theorem 4.1 (Proposition 6.48 in [PHP08])**.** *If (N, p, q, d) satisfies*

$$q > (6d+1) \cdot p,$$

then

$$D_{k_{\mathsf{priv}}} \left(E_{k_{\mathsf{pub}}}(m) \right) = m$$

for all $m \in R$.

Proof. Suppose $m \in \mathcal{M}$. Then since $k_{\mathsf{pub}} = [f]_q^{-1} \cdot [g]_q$, the NTRU ciphertext is of the form

$$[c]_q = [p \cdot k_{\mathsf{pub}} \cdot r + m]_q = [p]_q \cdot [f]_q^{-1} \cdot [g]_q \cdot [r]_q + [m]_q .$$

Multiplying both sides by $[f]_q$ and applying the distributive law gives

$$\begin{aligned} [f]_q \cdot [c]_q &= [f]_q \cdot \left([p]_q \cdot [f]_q^{-1} \cdot [g]_q \cdot [r]_q + [m]_q \right) \\ &= \left([f]_q \cdot [p]_q \cdot [f]_q^{-1} \cdot [g]_q \cdot [r]_q\right) + \left([f]_q \cdot [m]_q\right) \\ &= [1]_q \cdot [p \cdot g \cdot r + f \cdot m]_q \\ &= [p \cdot g \cdot r + f \cdot m]_q . \end{aligned}$$

But since $q > (6d + q)$, Lemma 4.2 implies

$$\mathrm{lift}_q \left([f]_q \cdot [c]_q \right) = p \cdot g \cdot r + f \cdot m.$$

Reducing modulo p and multiplying by both sides by the private polynomial $[f]_p^{-1}$ gives

$$\begin{aligned} [f]_p^{-1} \cdot \left([f]_p \cdot [c]_p \right) &= [f]_p^{-1} \cdot \left([p]_p \cdot [g]_p \cdot [r]_p + [m]_p \right) \\ &= \left([f]_p^{-1} \cdot [p]_p \cdot [g]_p \cdot [r]_p\right) + \left([f]_p^{-1} \cdot [f]_p \cdot [m]_p\right). \end{aligned}$$

As $[p]_p = [0]_p$ and $[f]_p \cdot [f]_p^{-1} = [1]_p$, we get

$$\left([f]_p^{-1} \cdot [p]_p \cdot [g]_p \cdot [r]_p\right) + \left([f]_p^{-1} \cdot [f]_p \cdot [m]_p\right) = [m]_p .$$

Since coefficient of m lies in $(\frac{p}{2}, \frac{p}{2})$, center lifting modulo p to $\mathbb{Z}[\zeta]$ gives the original plaintext. ∎

References

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A Ring-based Public Key Cryptosystem. In *Lecture Notes in Computer Science*, pages 267–288. Springer-Verlag, 1998.

[MCJ+16] Dustin Moody, Lily Chen, Stephen Jordan, Yi-Kai Liu, Daniel Smith, Ray Perlner, and Ren Peralta. NIST Report on Post-Quantum Cryptography, 04 2016.

[PHP08] Jill Pipher, Jeffrey Hoffstein, and Jill Pipher. Joseph H. Silverman An Introduction to Mathematical Cryptography, 2008.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, October 1997.

5 Appendix

SAGE Implementation 1: NTRUEncrypt

```
def Z(f):
    """Return polynomial in cyclotomic ring Z[zeta]"""
    Zx.<x> = PolynomialRing(ZZ, 'x')
    Z.<x> = Zx.quotient(x^N-1)
    return Z(f)

def Zmod(f, q, N):
    """Return polynomial in cyclotomic ring Zq[zeta]"""
    Zs.<x> = PolynomialRing(GF(q), 'x')
    Zmod.<x> = Zs.quotient(x^N-1)
    return Zmod(f)

def invmod(f, s, N):
    """Return modular inverse in cyclotomic ring Zq[zeta]"""
    Rs.<x> = PolynomialRing(GF(s),'x')
    return Rs(f).inverse_mod(x^N-1)

def lift(f, q):
    """Return center lift mod q of reduced cyclotomic polynomial"""
    for i in range(len(f)):
        f[i] = int(f[i]) % q
        if f[i] > (q / 2):
            f[i]-= q
    return R(f)

def keygen(N, p, q, d, f, g):
    """Generate NTRUEncrypt public-private keypair"""
    if d < 1:
        return "Invalid parameters: d must be a positive integer"
    if N < 3:
        return "Invalid parameters: N must be an odd prime"
    if q <= (6*d+1)*p:
        return "Invalid parameters: q <= (6*d+1)*p, decryption may fail
            "
    if gcd(p,q) != 1:
        return "Invalid parameters: p, q are not relatively prime"
    K_priv = [Z(f), invmod(f, p, N)]
    K_pub = invmod(f, q, N) * Zmod(g, q, N)
    return K_priv, K_pub

def encrypt(K_pub, r, N, q, m):
    """Return NTRUEncrypt ciphertext"""
    m, r, h = Zmod(m, q, N), Z(r), Zmod((K_pub).list(), q, N)
    return  ((p * h) * r) + m

def decrypt(N, p, q, d, K_priv, c):
    """Return decryption of NTRUEncrypt ciphertext"""
    a = lift((Zmod(K_priv, q, N) * c).list(), q)
    return lift((invmod(f, p, N) * a).list(), p)
```