Contents lists available at GrowingScience

International Journal of Data and Network Science

homepage: www.GrowingScience.com/ijds

A novel security analysis for a new NTRU variant with additional private key

Nurshamimi Salleh[a], Hailiza Kamarulhaili[a*] and Laith Abualigah[a,b,c,d,e,f]

[a]*School of Mathematical Sciences, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia*
[b]*Prince Hussein Bin Abdullah College for Information Technology, Al Al-Bayt University, Mafraq 130040, Jordan*
[c]*Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman 19328, Jordan*
[d]*Center for Engineering Application & Technology Solutions, Ho Chi Minh City Open University, Ho Chi Minh, Viet Nam*
[e]*Applied science research center, Applied science private university, Amman 11931, Jordan*
[f]*Faculty of Information Technology, Middle East University, Amman 11831, Jordan*

C H R O N I C L E

Article history:
Received: December 2, 2022
Received in revised format: December 29, 2022
Accepted: February 1, 2023
Available online: February 1, 2023

Keywords:
NTRU
Private Key
Public Key
Encryption
Decryption

A B S T R A C T

This paper proposes a new variant of NTRU with a slightly different critical formulation. The significance of this new variant is that it requires an additional private key to provide a tighter scheme. Because of these changes, modified key generation, encryption and decryption algorithms have been developed accordingly. The new variant is analyzed and tested against several well-known attacks, namely the alternate private key attack, brute force attack, meet-in-the-middle attack, multiple transmission attacks and lattice attack. Security properties related to these attacks have been established and explored to ensure the new variant is secure against the said attacks. Several examples are provided to illustrate the ideas.

1. Introduction

Due to the rapid development of mobile phone networks and the internet, security plays a crucial role in maintaining the secrecy of information for either people or organizations (Abu-Ulbeh et al., 2021; Mughaid et al., 2022). It spurs the rising demand for applications with optimum security since it will ensure the confidentiality of information transmissions in the communication network (Otair, Ibrahim, Abualigah, Altalhi, & Sumari, 2022). Intensive research in the cryptography field has suggested that the best candidate for secure applications with cryptographic techniques is public-key cryptography (Imam, Areeb, Alturki, & Anwer, 2021; Prasadh, Ramar, & Gnanajeyaraman, 2009). Public-key cryptography, sometimes known as asymmetric cryptography, is any cryptographic system with two cryptographic keys, an encryption key, and a decryption key (Otair et al., 2022). An encryption key usually is called a public key that has been shared publicly, whereas a decryption key usually is called a private key that has been kept secret. These cryptographic keys are distributed in such a way as to ensure that the contents of the message not be interpreted by unauthorized users, and the message can be transmitted by the cryptographic system securely where such a cryptographic system is so-called a public-key cryptosystem (Kalra & Sood, 2015; Xu, Dong, Ma, Liu, & Cliff, 2022). In public-key cryptography, the public-key cryptosystems are designed by manipulating the hardness of problems such as factorization problems, discrete logarithm problems, elliptic curves, discrete logarithm problems, and lattice problems (Huang, Zhou, Mi, Kuang, & Liu, 2022; Qin, Huang, & Fan, 2021). The reason is that the public key cryptosystems become infeasible to any practical means. Among these public key cryptosystems, the public key cryptosystem associated with the lattice problems is the hardest to break in practice. This type of public key cryptosystem is

* Corresponding author.
E-mail address: hailiza@usm.my (H. Kamarulhaili)

ISSN 2561-8156 (Online) - ISSN 2561-8148 (Print)

doi: 10.5267/j.ijdns.2023.2.001

constructed using the idea when the encryption is within a particular lattice, and then the decryption is solved based on some lattice problems (Alshurideh & Kurdi, 2023; Jarah, Jarrah, Almomani, AlJarrah, & Al-Rashdan, 2023). There are several lattice-based public key cryptosystems, including the AD (Ajtai-Dwork) cryptosystem (Ajtai & Dwork, 1997), GGH (Goldreich-Goldwasser-Halevi) cryptosystem (Goldreich, Goldwasser, & Halevi, 1997), NTRU (N-th degree Truncated polynomial Ring Unit) cryptosystem (Jeffery Hoffstein, 1996), and LWE (Learning with Errors) cryptosystem (Regev, 2009).

In order to offer effective protection via the Internet, security- and privacy-enhancing approaches are created (Kaaniche, Laurent, & Belguith, 2020). Using these methods, users should be able to communicate anonymously, whether by sending emails, making payments online, surfing the web, or posting to newsgroups. The most feasible method for obscuring messages and sender addresses is MixNet (Barsocchi et al., 2021). Numerous routing methods are contained in the anonymous channel and involve the transmission of anonymous data among nodes. This method of achieving anonymity is also the foundation of the MixNet structure. Following Chaumian MixNet's suggested work, several successful practical implementations of MixNet have been made thus far utilizing a variety of methodologies. In this paper (Ahmad, Kamal, Ahmad, Khari, & Crespo, 2021), the asymmetric NTRU cryptosystem-hybrid MixNet is proposed. They created a system for anonymous communication so that more individuals may participate in their safe conversation. It conceals how input and output are related in each Mix server step. Compared to Hybrid MixNet employing ElGamal and ECC, NTRU-based Hybrid MixNet performs better. According to the proposed system, Hybrid MixNet using NTRU operates on average 16.4 milliseconds faster than Hybrid MixNet utilizing ElGamal or ECC, which operate at 93.4 and 182.2 milliseconds faster, respectively.

Asymmetric Cryptography makes information unintelligible to an unauthorized user and provides confidentiality to genuine users. Encryption and decryption technology are solutions to protect data from unauthorized users. Many opportunistic network algorithms in the existing literature provide optimal performance. However, in this research work (Abouaroek & Ahmad, 2021), the NTRU post-quantum algorithm is proposed due to its high performance, low cost, and fast execution during the encryption and decryption of the data over the network. We also implemented and analyzed the performance of the proposed NTRU algorithm and compared its results with the Elliptic Curve Cryptography and ElGamal algorithm. After the analysis, they concluded that our proposed technique is highly effective and secure. In opportunistic networks, the nodes communicate wirelessly with one another and transfer data using the store-carry-forward method. Opportunistic networks have heterogeneous nodes with various characteristics, including high mobility, low power, low density, short radio range, and multiple security risks to unapproved nodes. To gain users' trust, the primary difficulty in an opportunistic network is to secure and safeguard the information during transmission. By merging the cryptographic methods that put the present world and the virtual world in a safer position, this problem is technically overcome.

The NTRU public key cryptosystem is based on efficient calculations with negligible storage and temporal complexity (Yassein, Al-Saidi, & Farhan, 2022). Many researchers were driven to enhance NTRU performance by substituting newly proposed algebraic structures for the original polynomial ring and mathematical structure (Gaubatz, Kaps, & Sunar, 2004). In this study (Salman & Yassein, 2022), a brand-new NTRU-analog cryptosystem dubbed QOBTRU is suggested, built on the Carternion algebra, a freshly created algebraic structure. In terms of computational and spatial complexity, QOBTRU is at least quadratically more complicated than the original NTRU. It aims to provide three selected highly performant multidimensional NTRU-like cryptosystems—QTRU, OTRU, and BITRU—with an alternate security and performance attribute. Comparing them to QOBTRU revealed its advantages over them. To show its effectiveness, detailed statistical and security analysis is carried out. By fusing the well-known NTRU public-key cryptosystem with the analytical solution of group rings, the authors of this study suggest two brand-new public-key cryptosystems (Mittal, Kumar, & Kumar, 2021). They discuss the security evaluation of these cryptosystems and demonstrate how much more secure they are than the NTRU public-key cryptosystem. More specifically, they demonstrate that our new cryptosystems' security depends on resolving several challenging problems just recently identified, such as inverse computation difficulties and discrete logarithm problems in group rings. Talk about two instances to demonstrate how both cryptosystems' encryption and decryption processes work. NTRU cryptosystem is an encryption algorithm that interprets the lattice in the form of convolution polynomial rings. Specifically, the ring of convolution (or truncated) polynomials of degree $N - 1$ with the integer coefficients defined as the quotient ring, which is as follows.

$$R = \frac{\mathbb{Z}[X]}{X^N - 1}. \tag{1}$$

For moduli p and q, the ring of convolution (or truncated) polynomial N - 1 with the integer coefficients are defined as the quotient ring, which is as follows.

$$R_p = \frac{(\mathbb{Z}/p\mathbb{Z})[X]}{X^N - 1}, \tag{2}$$

$$R_q = \frac{(\mathbb{Z}/q\mathbb{Z})[X]}{X^N - 1}, \tag{3}$$

1.1 Our Contribution

This paper proposes an improved NTRU cryptosystem associated with the new private keys in the new spaces from the original NTRU cryptosystem. We use new private keys to generate a new public key, and modify the encryption and decryption to retrieve the message as the NTRU cryptosystem. Furthermore, we analyze the security of the proposed scheme by highlighting that our proposed scheme is secure against known attacks, such as the alternate private key attack, brute force attack, meet-in-the-middle attack, multiple transmission attack, and lattice attack. In addition, we compare the speed of the proposed scheme with the NTRU cryptosystem using the computational complexity based on the arithmetic operations. We also compare the security level of the private key and message for the proposed scheme and the NTRU cryptosystem.

This paper is organized as follows: Section 2 gives an overview of NTRU cryptosystem and some fundamental concepts of the lattices. Section 3 introduces the proposed scheme, NTRU, with modified key generation, encryption, and decryption. Section 4 discusses the security analysis for some attacks on the proposed scheme and Section 5 compares the proposed scheme and the NTRU cryptosystem. Finally, Section 6 presents the conclusion.

2. Materials and methods

This section gives an overview of NTRU cryptosystem and recalls some fundamental knowledge on the short vectors in lattices that are required by some results in this paper.

2.1 NTRU Cryptosystem

Consider the following parameters and spaces.

Table 1
Parameters for NTRU

Parameter	Description
N	The dimension of R
p	The small modulus to which each reduced coefficient
q	The large modulus to which each reduced coefficient
d_f	The number of coefficients of polynomial f
d_g	The number of coefficients of polynomial g
d	The number of coefficients of polynomial ϕ

Table 2
Spaces for NTRU

Space	Description
$\mathcal{L}_f$	The set of polynomials in R having d_f 1s and d_f-1 -1s.
$\mathcal{L}_g$	The set of polynomials in R having d_g 1s and d_g -1s.
$\mathcal{L}_\phi$	The set of polynomials in R having d 1s and d -1s.
$\mathcal{L}_m$	The set of polynomials in R having coefficients lying between $-\frac{1}{2}(p-1)$ and $\frac{1}{2}(p-1)$ if p is odd or coefficients lying between $-\frac{p}{2}$ and $\frac{p}{2}$ if p is even.

The NTRU cryptosystem is constructed as follows (Sever & Özdemir, 2021).

Key Generation

i. Choose a polynomial $f(X) \in \mathcal{L}_f$ such that f is invertible and satisfying the following:
$$f(X) * f_q^{-1}(X) \equiv 1 \pmod{q} \quad (4)$$
$$f(X) * f_p^{-1}(X) \equiv 1 \pmod{p}. \quad (5)$$
ii. Choose a polynomial $g(X) \in \mathcal{L}_g$.
iii. Compute a polynomial $h(X) = f_q^{-1}(X) * g(X) \pmod{q}$.

The private key is the pair (f, f_p^{-1}) and the public key is a polynomial h.

Encryption

Bob wants to send a message to Alice. He converts the message into the encrypted message using Alice's public key.

i. Represent plaintext as a polynomial $m(X) \in \mathcal{L}_m$.
ii. Choose randomly a polynomial $\phi(X) \in \mathcal{L}_\phi$.
iii. Compute ciphertext as a polynomial $e(X) \equiv p\phi(X) * h(X) + m(X) \pmod{q}$.

Then Bob sends the ciphertext e to Alice.

Decryption

Alice wants to decrypt the encrypted message received from Bob. She retrieves message from the encrypted message using her private key as follows.

i. Compute temporary polynomial $a(X) \equiv f(X) * e(X) \pmod{q}$.
ii. Center-lift a from R_q to R.
iii. Compute polynomial $b(X) \equiv f_p^{-1}(X) * a(X) \pmod{p}$.
iv. Center-lift b from R_p to R.

Then Alice retrieves the message.

2.2 Gaussian Expected Shortest Length

The following Gaussian heuristic is required to compute the length of a shortest nonzero vector in a lattice $\mathcal{L}$.

Definition 1: *Let* $\mathcal{L}$ be a lattice of dimension n (Jeffrey Hoffstein, Pipher, Silverman, & Silverman, 2008). The Gaussian expected shortest length is calculated as follows.

$$\sigma(\mathcal{L}) = \sqrt{\frac{n}{2\pi e}}(\det \mathcal{L})^{1/\mathrm{n}}. \tag{6}$$

The Gaussian heuristic says that a shortest nonzero vector in a "randomly chosen lattice" will satisfy, which is presented as follows.

$$\|v_{shortest}\| \approx \sigma(\mathcal{L}). \tag{7}$$

2.3 Hermite Theorem

The following Hermite theorem is required to estimate the length of a shortest nonzero vector in a lattice $\mathcal{L}$ (Jeffrey Hoffstein et al., 2008).

Theorem 1: *Every lattice* $\mathcal{L}$ *of dimension n contains a nonzero vector* $v \in \mathcal{L}$ *satisfying*

$$\|v\| \leq \sqrt{n}\,(\det(\mathcal{L}))^{1/n}.$$

3. The proposed method

Remarkably, all private keys in the proposed scheme chosen to be invertible in R_q (or R_p). This condition is needed in generating the public key and the decryption process. Thus, all private keys must not be in $\mathcal{L}(d_1, d_2)$ because the elements in $\mathcal{L}(d_1, d_2)$ never have inverses in R_q (or R_p). The following result will show that elements in $\mathcal{L}(d_1, d_2)$ is not invertible in R_q (or R_p).

Proposition 1 *For any positive integers* d_1 *and* d_2*, let* $\mathcal{L}(d_1, d_2) \subset (\mathbb{Z}/q\mathbb{Z})[X]$ *where* q *is a prime. If* $f(x)$ in $\mathcal{L}(d_1, d_2)$, then $f(1) \equiv 0 \pmod{q}$. *Therefore,* $f(x)$ *is not invertible in* R_q.

Proof. Suppose that $f(x)$ in $\mathcal{L}(d_1, d_2)$ and $x - 1$ is a factor of $f(x)$. Then $f(x) = (x-1)a(x)$ for some $a(x)$. Thus, $f(1) \equiv (1-1)a(1) \pmod{q} = 0 \pmod{q}$,
where 1 is a root of $f(x)$.

It is known that $f(x)$ in $(\mathbb{Z}/q\mathbb{Z})[X]$ has an inverse if $f(x)$ is coprime to q. By definition, $f(x)$ and q are coprime if $\gcd(f(x), q) = 1$. So, there are $a(x), b(x)$ in $(\mathbb{Z}/q\mathbb{Z})[X]$ such that $f(x)a(x) + qb(x) = 1$.

Then, $f(x)a(x) = 1 - qb(x)$, that is, $f(x)a(x) = 1 \pmod{q}$. Use the fact that 1 is a root of $f(x)$ in the latter equation yields $0 \pmod{q} \equiv f(1)a(1) \equiv 1 \pmod{q}$.

This contradiction shows that $f(x)$ is not coprime to q, and $f(x)$ in $\mathcal{L}(d_1, d_2)$ has no inverse in $(\mathbb{Z}/q\mathbb{Z})[X]$. Therefore, $f(x)$ is not invertible in R_q.

Furthermore, there is an important method is applied on the polynomials in the decryption process. This method is used to lifting the coefficients of polynomial in R_q (or R_p) to R as follows.

Proposition 2 *Let $c(x)$ is in R_q and $c'(x)$ is the center-lift of $c(x)$ to R. If any coefficients c_i within the interval $-\frac{q}{2} < c_i \leq \frac{q}{2}$, then $c_i' = c_i$. For any coefficient $c_i < -\frac{q}{2}$, then $c_i' = c_i + q$. For any coefficient $c_i > \frac{q}{2}$, then $c_i' = c_i - q$. Therefore, all the coefficients of $c'(x)$ are in the interval $-\frac{q}{2} < c_i' \leq \frac{q}{2}$.*

Proof. Let $c(x)$ has a representation of the form $\sum_{i=0}^{N-1} c_i x^i \mod q$. Center-lifting $c(x)$ yields $c'(x)$ with all the coefficients of $c(x)$ are chosen within the interval $-\frac{q}{2} < c'_i \leq \frac{q}{2}$. For any coefficient c_i that already in the interval $-\frac{q}{2} < c'_i \leq \frac{q}{2}$, the coefficients of $c'(x)$ are exactly in the interval $-\frac{q}{2} < c'_i \leq \frac{q}{2}$.

For any coefficient $c_i < -\frac{q}{2}$, it shows that the coefficient c_i is not in the interval $-\frac{q}{2} < c'_i \leq \frac{q}{2}$.
To ensure the coefficients c_i is in the interval $-\frac{q}{2} < c'_i \leq \frac{q}{2}$, let calculate the difference from $-\frac{q}{2}$ to c_i, and then minus $\frac{q}{2}$ with the difference from $-\frac{q}{2}$ to c_i. Thus, the coefficients of $c'(x)$ is $\frac{q}{2} - \left(\left(-\frac{q}{2}\right) - c_i\right)$. Simplifying this yield $c_i' = c_i + q$.

Similarly, for any coefficient $c_i > \frac{q}{2}$, it also shows that the coefficient c_i is not in the interval $-\frac{q}{2} < c'_i \leq \frac{q}{2}$. To ensure the coefficients c_i is in the interval $-\frac{q}{2} < c'_i \leq \frac{q}{2}$, let calculate the difference from c_i to $\frac{q}{2}$, and then add $-\frac{q}{2}$ with the difference from c_i to $\frac{q}{2}$. Thus, the coefficients of $c'(x)$ is $\left(-\frac{q}{2}\right) + \left(c_i - \frac{q}{2}\right)$. Simplifying this yield $c_i' = c_i - q$. Now, all the coefficients of $c'(x)$ are in the interval $-\frac{q}{2} < c'_i \leq \frac{q}{2}$.

3.1 The Construction of the Proposed Scheme

This proposed scheme assumes the same parameters as per table 1 in Section 2.1 except for the parameter d_g. There are a few new parameters such as the parameters d_r and d_s. This proposed scheme also assumes the same spaces as per table 2 in Section 2.1 except for the space $\mathcal{L}_g$. There are a few new spaces, including $\mathcal{L}_r = \mathcal{L}(d_r, d_r + 1)$, and $\mathcal{L}_s = \mathcal{L}(d_s + 1, d_s)$. Therefore, the proposed scheme is constructed as follows.

Key Generation

i. Choose the polynomial $f(X) \in \mathcal{L}_f$ such that f is invertible and satisfying $f(X) * f_q^{-1}(X) \equiv 1 \pmod q$,
ii. Choose the polynomial $r(X) \in \mathcal{L}_r$ such that r is invertible and satisfying $r(X) * r_p^{-1}(X) \equiv 1 \pmod p$,
iii. Choose the polynomial $s(X) \in \mathcal{L}_s$ such that s is invertible and satisfying $s(X) * s_q^{-1}(X) \equiv 1 \pmod q$,
iv. Compute a polynomial $\hbar(X) = f_q^{-1}(X) * r(X) * s_q^{-1}(X) \pmod q$.

The private key is the triple (f, s, r_p^{-1}) and the public key is the polynomial $\hbar$.

Encryption

Bob wants to send a message to Alice. He converts the message becomes the encrypted message using Alice's public key as follows.

i. Represent plaintext as a polynomial $m(X) \in \mathcal{L}_m$.
ii. Choose randomly a polynomial $\phi(X) \in \mathcal{L}_\phi$.
iii. Compute ciphertext as a polynomial $e(X) \equiv \left(p\phi(X) + m(X)\right) * \hbar(X) \pmod q$

Then Bob sends the ciphertext e to Alice.

Decryption

Alice wants to decrypt the encrypted message received from Bob. She retrieves message from the encrypted message using her private keys as follows.

i. Compute a polynomial $a(X) \equiv f(X) * s(X) * e(X) \pmod q$
ii. Center-lift a from R_q to R.
iii. Compute a polynomial $b(X) \equiv r_p^{-1}(X) * a(X) \pmod p$
iv. Center- lift b from R_p to R.

Then Alice retrieves the message.

Proof of Correctness

Proposition 3 *The proposed decryption scheme is correct.*

Proof. To decrypt the ciphertext decrypt the ciphertext e to plaintext m, the private keys f, r and s will be used. Observe the temporary polynomial a defined as follows.

$$a(X) \equiv f(X) * s(X) * e(X) \pmod q \tag{8}$$

Using the encrypted message $e(X) = \big(p\phi(X) + m(X)\big) * h(X)$ with the public key $h(X) \equiv f_q^{-1}(X) * r(X) * s_q^{-1}(X)$ in (1) gives $a(X) \equiv f(X) * s(X) * \big(p\phi(X) + m(X)\big) * f_q^{-1}(X) * r(X) * s_q^{-1}(X) \pmod q$. It is known that private keys f and s satisfied the condition $f(X) * f_q^{-1}(X) \equiv 1 \pmod q$ and $s(X) * s_q^{-1}(X) \equiv 1 \pmod q$ respectively. Now, the latter inequality becomes as follows.

$$a(X) \equiv \big(p\phi(X) + m(X)\big) * r(X) \pmod q \tag{9}$$

Since the parameter p is a small modulus, and the polynomial ϕ, m and r all have small coefficients compare to q, then all coefficients of $\big(p\phi(X) + m(X)\big) * r(X)$ lies in the interval $(-q/2, q/2]$.

To retrieve the message m, multiply the inverse of private key r with Eq. (9) produces $b(X) \equiv p\phi(X) + m(X) \pmod p$ since the private keys r is invertible and satisfies the condition $r(X) * r_p^{-1}(X) \equiv 1 \pmod p$. Then, solve a polynomial b in modulo p yields.

$$b(X) \equiv m(X) \pmod p \tag{10}$$

Because $p\phi(X) \equiv 0 \pmod p$. From eq. (10), it shown that the polynomial m is in modulo p and this implies that all coefficients of m lies in the interval $(-p/2, p/2]$. Therefore, the plaintext m is exactly retrieved.

Proposition 4 *If the proposed parameter* (p, q, d, d_r) *are chosen to satisfy* $(2d + 4d_r + 3)p < q$*, then the polynomial* $a(X)$ *computed by Alice is equal to Bob's plaintext* $m(X)$*.*

Proof. First let find out the shape of Alice's basis calculation of $a(X)$, that is, $a(X) \equiv f(X) * s(X) * e(X) \pmod q \equiv \big(p\phi(X) + m(X)\big) * r(X) \pmod q$ and $\equiv p\phi(X) * r(X) + m(X) * r(X) \pmod q$, which shows that the polynomial $p\phi(X) * r(X) + m(X) * r(X)$ is in R_q.

This implies that the magnitude of the polynomial $p\phi(X) * r(X) + m(X) * r(X)$ is strictly less than $q/2$ where all the individual coefficient in the polynomial $p\phi(X) * r(X) + m(X) * r(X)$ have their own magnitudes largest possible coefficient.

Now, consider the polynomial $p\phi(X) * r(X) + m(X) * r(X)$ is in R and let check the bound for its individual largest possible coefficient. First, the largest possible coefficient of $\phi(X) * r(X)$ is $d + d_r + 1$ since the polynomial $\phi(X)$ is in $\mathcal{L}(d, d)$ and $r(X)$ is in $\mathcal{L}(d_r, d_r + 1)$.

Next, the largest possible coefficient of $m(X) * r(X)$ is $\frac{p}{2} \cdot (2d_r + 1)$ since the polynomial $m(X)$ is in $\mathcal{L}_m$ and $r(X)$ is in $\mathcal{L}(d_r, d_r + 1)$. To ensure the polynomial $p\phi(X) * r(X) + m(X) * r(X)$ computed exactly in R, all coefficients of the polynomial $p\phi(X) * r(X) + m(X) * r(X)$ is strictly smaller than $q/2$. In other words, the magnitude of all coefficients of the polynomial $p\phi(X) * r(X) + m(X) * r(X)$ is

$p \cdot (d + d_r + 1) + \frac{p}{2} \cdot (2d_r + 1) < \frac{q}{2}$ or $(2d + 4d_r + 3)p < q$. This condition ensures the polynomial $p\phi(X) * r(X) + m(X) * r(X)$ or the polynomial $a(X)$ computed by Alice is exactly in R.

Hence, Alice computes the polynomial $b(X)$ modulo p yields $b(X) \equiv m(X) \pmod p$. Since the polynomial $m(X)$ is in R_p, then by Proposition 2, Alice will be able to obtain the polynomial $m(X)$ exactly in R, in which is the same as Bob's plaintext $m(X)$ in R. Therefore, the polynomial $a(X)$ computed by Alice is equal to Bob's plaintext $m(X)$.

Numerical Illustration of the Proposed Scheme

Consider the parameter $(N, p, q, d_f, d_r, d_s, d) = (7,3,47,2,2,2,2)$ and the following polynomials $f(X) = X^4 - X^2 + 1 \in \mathcal{L}(2,1), r(X) = X^5 + X^4 - X^2 - X - 1 \in \mathcal{L}(2,3)$,

$s(X) = X^6 - X^4 + X^2 + X - 1 \in \mathcal{L}(3,2)$, The decryption will work since $45 = (2d + 4d_r + 3)p < q = 47$. Then Bob computes the inverses of the private key f, r and s, and the public key h as follows.

$$f_q^{-1}(X) = 46X^6 + X^5 + 46X^4 + X^3 + 1 \in R_{47}, \quad (11)$$
$$r_p^{-1}(X) = X^6 + 2X^5 + 2 \in R_3, \quad (12)$$
$$s_q^{-1}(X) = 10X^6 + 46X^5 + 33X^4 + 12X^3 + 27X^2 + 9X + 5 \in R_{47}, \quad (13)$$
$$\mathcal{h}(X) \equiv 7X^6 + 18X^5 + 10X^4 + 19X^3 + 44X^2 + 18X + 24 \pmod{47} \quad (14)$$

Then, Alice chooses message m as follows.

$$m(X) = X^5 + X^4 - X^3 - X + 1 \in \mathcal{L}_m\,, \quad (15)$$

and the random polynomial ϕ as follows.

$$\phi(X) = X^6 + X^5 - X^3 - 1 \in \mathcal{L}(2,2). \quad (16)$$

By using those together with the public key h, she calculates an encrypted message e as

$e(X) \equiv 33X^6 + 7X^5 + 42X^4 + 25X^3 + 28X^2 + 39X + 13 \pmod{47}$. Then she sends the encrypted message e to Bob.

Now, Bob receives the encrypted message e from Alice. Next, Bob calculates the temporary polynomial a using a private key f and s. The calculation of a yields $a(X) \equiv 38X^6 + 43X^5 + 4X^4 + 12X^3 + 8X^2 + 44X + 38 \pmod{47}$ and center-lifting it modulo 47 with the coefficients are chosen from $\{-23, -22, \ldots, 22, 23\}$ gives $a(X) = -9X^6 - 4X^5 + 4X^4 + 12X^3 + 8X^2 - 3X - 9$. Finally, Bob can retrieve the message m by multiplying the inverse r with a as follows.

$$r_p^{-1}(X) * a(X) \equiv X^5 + X^4 + 2X^3 + 2X + 1 \pmod{3} \quad (17)$$

and then center-lifting it modulo 3 with the coefficients are chosen from $\{-1,0,1\}$ as follows.

$$X^5 + X^4 - X^3 - X + 1 = m(X). \quad (18)$$

3.2 The Algorithm of the Proposed Scheme

The following algorithms illustrated the key generation, encryption, and decryption of the proposed scheme.

Algorithm 1 Keys Generation

Input: $N, p, q, d_f, d_r, d_s \in N$
Output: Private key f, Private key r, Private key s, Public key $\mathcal{h}$

repeat
  $f \leftarrow \mathcal{L}(d_f, d_f - 1)$
until f is invertible in modulo q
repeat
  $r \leftarrow \mathcal{L}(d_r, d_r + 1)$
until r is invertible in modulo p
repeat
  $s \leftarrow \mathcal{L}(d_s + 1, d_s)$
until s is invertible in modulo q
Compute
$\mathcal{h}(X) \equiv f_q^{-1}(X) * r(X) * s_q^{-1}(X) \pmod q$
return Private key $\leftarrow f$, Private key $\leftarrow r$, Private key $\leftarrow s$, Public key $\leftarrow \mathcal{h}$

Algorithm 2 Encryption

Input: $p, q, d \in N$, Public key $\hbar$

Output: Ciphertext e

repeat
  $m \leftarrow \mathcal{L}_m$
until the coefficients of m are $-\frac{1}{2}(p-1)$, $\frac{1}{2}(p-1)$ or 0
repeat
  $\phi \leftarrow \mathcal{L}(d, d)$
until the number of $+1$ and -1 in ϕ are each equal to d
Compute
  $e(X) \equiv \big(p\phi(X) + m(X)\big) * \hbar(X) \pmod{q}$
return Ciphertext $\leftarrow e$

Algorithm 3 Decryption

Input: $p, q \in N$, Private key f, Private key r, Private key s, Ciphertext e

Output: Plaintext (or message) m

Compute $a(X) \equiv f(X) * s(X) * e(X) \pmod{X^N - 1, q}$
Compute $a'(X)$ = Center-lift of $a(X)$
for $i = 0$ to $N - 1$ **do**
  $a'(X)$ where its coefficients within the interval: $-\frac{q}{2} < a'_i \leq \frac{q}{2}$
  if the coefficient $a_i < -\frac{q}{2}$ **then**
    $a'_i = a_i + q$
  else if the coefficient $a_i > \frac{q}{2}$ **then**
    $a'_i = a_i - q$
  else
    $a'_i = a_i$
  end if
end for
Compute $b(X) = r_p^{-1}(X) * a'(X) \pmod{p}$
Compute $b'(X)$ =Center-lift of $b(X)$
for $i = 0$ to N-1 **do**
  $b'(X)$ where its coefficients within the interval: $-\frac{p}{2} < b'_i \leq \frac{p}{2}$
  if the coefficient $b_i < -\frac{p}{2}$ **then**
    $b'_i = b_i + p$
  else if $b_i > \frac{p}{2}$ **then**
    $b'_i = b_i - p$
  else
    $b'_i = b_i$
  end if
end for
Obtain $b'(X) = m(X)$
return Plaintext $\leftarrow m$

4. Security analysis

The security of the proposed scheme depends on well-known attacks such as the alternate keys attack, brute force attack, meet-in-the-middle attack, multiple transmission attacks and lattice attack. Note that the relationship $f(X) * s(X) * \hbar(X) \equiv r(X) \pmod{q}$, will be used in most attacks against the proposed scheme.

4.1 Alternate Private Key Attack

The alternate private keys attack is introduced by (Al-Saidi & Yassein, 2017), and this attack can find an alternate private key that has the same characteristics as the private key to decrypt the same message. The following definition defines an alternate private key.

Definition 2: *An alternate private key, denoted as f' is a rotation of the private key f, that is $f' = X^i * f(X)$ for some positive integer i.*

The above definition will be used in the result below.

Theorem 2. *The proposed scheme is secure against alternate private keys attack.*

Proof. By Definition 2, let $f'(X) = X^i * f(X)$ is any rotation of the private key f and $s'(X) = X^i * s(X)$ is any rotation of the private key s. It is known that $f(X) * s(X) * h(X) \equiv r(X) \pmod q$, then $X^i * \big(f(X) * s(X) * h(X)\big) \equiv X^i * r(X) \pmod q$.

Let $r'(X) = X^i * r(X)$ be any corresponding rotation of private key r. Thus, the latter inequality be expressed as $f'(X) * s(X) * h(X) \equiv r'(X) \pmod q$. Again, $X^i * \big(f'(X) * s(X) * h(X)\big) \equiv X^i * r'(X) \pmod q$ and this gives $f'(X) * s'(X) * h(X) \equiv X^i * r'(X) \pmod q$.

Assume that a' be an alternate temporary polynomial a, that is, $a'(X) = X^i * a(X)$. To decrypt the message, calculate $a'(X) \equiv f'(X) * s'(X) * e(X) \pmod q$, then

$$a'(X) \equiv X^i * f(X) * X^i * s(X) * \big(p\phi(X) + m(X)\big) * h(X) \pmod q$$

$$\equiv X^{2i} * f(X) * s(X) * \big(p\phi(X) + m(X)\big) * f_q^{-1}(X) * s_q^{-1}(X) * r(X) \pmod q$$

Using $f(X) * f_q^{-1}(X) \equiv 1 \pmod q$ and $s(X) * s_q^{-1}(X) \equiv 1 \pmod q$ in the above equation yields,

$$a'(X) \equiv X^{2i} * \big(p\phi(X) * r(X) + m(X) * r(X)\big) \pmod q \equiv X^{2i} * a(X) \pmod q.$$

This contradicts the assumption that $a'(X) = X^i * a(X)$, and this shows that a' is not an alternate temporary polynomial a. Hence, this concludes that f' and s' cannot be used to decrypt the same messages as the private keys f and s. Therefore, the proposed scheme is secure against alternate private keys attack.

4.2 Brute Force Attack

The brute force attack is proposed by (Jeffrey Hoffstein, Pipher, & Silverman, 1998), and this attack against the proposed scheme uses the number of elements in the search spaces to find the private keys and the message. The following definition defines the number of elements in the search space that will be used in this brute force attack.

Definition 3: *Let N be the degree of polynomial, d_1 be the numbers of coefficients equal to 1, and d_2 be the numbers of coefficients equal to -1. Then the number of elements in the search space is defined by the following equation.*

$$\#\mathcal{L}(d_1, d_2) = \binom{N}{d_1}\binom{N - d_1}{d_2} \tag{19}$$

where N, d_1, and d_2 are some positive integers.

In the NTRU cryptosystem, $\#\mathcal{L}_g$ has been used to determine its key security instead of $\#\mathcal{L}_f$ because space $\mathcal{L}_g$ is smaller than the space $\mathcal{L}_f$. For the same reason, the proposed scheme will use $\#\mathcal{L}_f$ to determine the key security instead of $\#\mathcal{L}_r$ and $\#\mathcal{L}_s$, and will use $\#\mathcal{L}_\phi$ to determine the message security.

Theorem 3: *The proposed scheme is secure against brute force attack.*

Proof. Assume that an attacker who knows the public key h can find the private keys and an attacker who knows the encrypted message e can recover the message.

To find the private keys, an attacker can check $f(X) * s(X) * h(X) \pmod q$ has small coefficients compared to q or not by trying all possible $f \in \mathcal{L}\big(d_f, d_f - 1\big)$ and all possible $s \in \mathcal{L}(d_s + 1, d_s)$. Or an attacker can check $r(X) * f^{-1}(X) * h^{-1}(X) \pmod q$ has small coefficients compared to q or not by trying all possible $r \in \mathcal{L}(d_r, d_r + 1)$ and also all possible $f \in \mathcal{L}\big(d_f, d_f - 1\big)$. Or an attacker can check $r(X) * s^{-1}(X) * h^{-1}(X) \pmod q$ has small coefficients compared to q or not by trying all possible $r \in \mathcal{L}(d_r, d_r + 1)$ and also all possible $s \in \mathcal{L}(d_s + 1, d_s)$.

Thus, the number of elements in the search space for the private keys f, s and r are given by $\#\mathcal{L}_f = \#\mathcal{L}(d_f, d_f - 1) = \binom{N}{d_f}\binom{N-d_f}{d_f-1} = \frac{N!}{(d_f)(N-2d_f+1)!\left((d_f-1)!\right)^2}$, $\#\mathcal{L}_s = \#\mathcal{L}(d_s+1, d_s) = \binom{N}{d_s+1}\binom{N-d_s-1}{d_s} = \frac{N!}{(d_s+1)(N-2d_s-1)!(d_s!)^2}$, and $\#\mathcal{L}_r = \#\mathcal{L}(d_r, d_r+1) = \binom{N}{d_r}\binom{N-d_r}{d_r+1} = \frac{N!}{(d_r+1)(N-2d_r-1)!(d_r!)^2}$, respectively. Since the space $\mathcal{L}_f$ is smaller than the space $\mathcal{L}_s$ and $\mathcal{L}_r$, then $\#\mathcal{L}_f$ will be used to determine the key security.

To recover the message, an attacker can check $e(X) * \hbar^{-1}(X) - p\phi(X) \ (mod\ q)$ has small coefficients compared to q or not by trying all possible $\phi \in \mathcal{L}(d, d)$. Thus, the number of elements in the search space for the random polynomial ϕ is given by $\#\mathcal{L}_\phi = \#\mathcal{L}(d, d) = \binom{N}{d}\binom{N-d}{d} = \frac{N!}{(N-2d)!(d!)^2}$, and this $\#\mathcal{L}_\phi$ will be used to determine the message security.

From the fact above, $\#\mathcal{L}_f$ will represent the search time for finding the private keys and $\#\mathcal{L}_\phi$ will represent the search time for finding the message. For sufficiently large N, $\#\mathcal{L}_f$ and $\#\mathcal{L}_\phi$ will produce a considerably long search time for finding the private keys and recovering the message respectively. When the search time is elongate, the brute force attack is difficult and infeasible to find the private key and recover the message, and this contradict our assumption. Therefore, the proposed scheme is secure against brute force attack.

The following table illustrates the comparison between the NTRU cryptosystem and the proposed scheme based on the list parameters used in (Jeffrey Hoffstein et al., 1998).

Table 3
The Number of Elements in The Search Space for The Private Keys and The Message.

N	q	d_f	d_g	d	NTRU [HPS 98]		Proposed scheme	
					$\#\mathcal{L}_g$	$\#\mathcal{L}_\phi$	$\#\mathcal{L}_f$	$\#\mathcal{L}_\phi$
107	64	15	12	5	1.4×10^{30}	8.9×10^{15}	9.5×10^{33}	8.9×10^{15}
167	128	61	20	18	8.4×10^{49}	4.3×10^{46}	6.4×10^{76}	4.3×10^{46}
503	256	216	72	55	3.7×10^{171}	2.3×10^{145}	5.4×10^{216}	2.3×10^{145}

4.3 Meet-In-The-Middle Attack

The meet-in-the-middle attack was first investigated by (Howgrave-Graham, 2007) based on the original meet-in-the-middle attack by Andrew Odlyzko, which applied to an encrypted message e against a random polynomial ϕ. Then, the NTRU cryptosystem applied a similar attack as in (van Vredendaal, 2016) on the private key f. Here, the meet-in-the-middle-attack is applied to the private keys f and s.

The following steps described the meet-in-the-middle attack for the proposed scheme.

1) Enumerate the polynomial $\left(f_1(X) * s_1(X) + f_2(X) * s_2(X)\right)$.
Assume that $N > k$ where N is odd number and k is a positive integer. Let $f(X) = f_1(X) + f_2(X)$ or $f(X) = \sum_{i=0}^{(N+1)/2-1} a_i X^i + \sum_{i=(N+1)/2}^{N-1} a_i X^i$, and $s(X) = s_1(X) + s_2(X)$ or $s(X) = \sum_{i=0}^{(N+1)/2-1} b_i X^i + \sum_{i=(N+1)/2}^{N-1} b_i X^i$, respectively.

   Compute $(f_1(X) * s_1(X) + f_2(X) * s_2(X)) * \hbar(X) \ (mod\ q)$ and convert into a binary representation based on the most significant bit of the first k coordinates of $(f_1(X) * s_1(X) + f_2(X) * s_2(X)) * \hbar(X) \ (mod\ q)$. To obtain the binary representation of the first k coefficients of $(f_1(X) * s_1(X) + f_2(X) * s_2(X)) * \hbar(X) \ (mod\ q)$, take each of such first k coefficients and convert it into binary numbers. Then take the leading bit of each binary number to form the bin labeled for the polynomial $(f_1(X) * s_1(X) + f_2(X) * s_2(X)) * \hbar(X) \ (mod\ q)$.

2) Enumerate the polynomial $\left(f_1(X) * s_2(X) + f_2(X) * s_1(X)\right)$. Compute $-(f_1(X) * s_2(X) + f_2(X) * s_1(X)) * \hbar(X) \ (mod\ q)$ and convert $-(f_1(X) * s_2(X) + f_2(X) * s_1(X)) * \hbar(X) \ (mod\ q)$ into a binary representation based on the most significant bit of the first k coordinates of $-\left(f_1(X) * s_2(X) + f_2(X) * s_1(X)\right) * \hbar(X) \ (mod\ q)$ as well as a binary representation based on the most significant bit of the first k coordinates of $-(f_1(X) * s_2(X) + f_2(X) * s_1(X)) * \hbar(X) \ (mod\ q)$ with addition 1s and -1s on it. To obtain these binary representations, take each such first k coefficients and converts it to binary number. Then take the leading bit of each binary number to form the bin labeled for the polynomial $-(f_1(X) * s_2(X) + f_2(X) * s_1(X)) * \hbar(X) \ (mod\ q)$. Next, take the leading bit of each binary number which has been added 1s and -1s on it to form the list of the bin labeled for the polynomial $-(f_1(X) * s_2(X) + f_2(X) * s_1(X)) * \hbar(X) \ (mod\ q)$.

Check the matches for the bin labeled for the polynomial $(f_1(X) * s_1(X) + f_2(X) * s_2(X)) * h(X) \pmod q$ against the bin labeled for the polynomial $-(f_1(X) * s_2(X) + f_2(X) * s_1(X)) * h(X) \pmod q$ together with all the lists of the bin labeled for the polynomial $-(f_1(X) * s_2(X) + f_2(X) * s_1(X)) * h(X) \pmod q$. If the matches having the same bit, then $\big(f_1(X) + f_2(X)\big) * \big(s_1(X) + s_2(X)\big)$ can be the private keys.

Theorem 4: *The proposed scheme is secure against meet-in-the-middle attack.*

Proof. Assume that the meet-in-the-middle attacks can be used to find the private key f and s. Let the private key f and s defined by $f(X) = f_1(X) + f_2(X)$ and $s(X) = s_1(X) + s_2(X)$ respectively. It is known that $f(X) * s(X) * h(X) \equiv r(X) \pmod q$, then $(f_1(X) + f_2(X)) * (s_1(X) + s_2(X)) * h(X) \equiv r(X) \pmod q$. Expanding the above equation yields $\big(f_1(X) * s_1(X) + f_2(X) * s_2(X)\big) * h(X) \equiv r(X) - \big(f_1(X) * s_2(X) + f_2(X) * s_1(X)\big) * h(X) \pmod q$. Since r is ternary polynomial, this means that the coefficients of $(f_1(X) * s_1(X) + f_2(X) * s_2(X)) * h(X) \pmod q$ and $-(f_1(X) * s_2(X) + f_2(X) * s_1(X)) * h(X) \pmod q$ will be either $-1, 0$ or 1 modulo q. Using the meet-in-the middle attack, obtain the matches $(f_1(X) * s_1(X) + f_2(X) * s_2(X)) * h(X) \pmod q$ against $-(f_1(X) * s_2(X) + f_2(X) * s_1(X)) * h(X) \pmod q$. When the matches have the same bit, it shows that the inequalities $\big(f_1(X) + f_2(X)\big) * \big(s_1(X) + s_2(X)\big) * h(X) \pmod q$, has small coefficients and thus $\big(f_1(X) + f_2(X)\big) * \big(s_1(X) + s_2(X)\big)$ can be a private key. For a sufficiently large N, it is difficult to obtain the same bit for $\big(f_1(X) * s_1(X) + f_2(X) * s_2(X)\big)$. Thus, the private keys f and s cannot be found and this contradict our assumption that the meet-in-the middle attack would be able to find the private keys f and s. Therefore, the proposed scheme is secure against meet-in-the-middle attack.

In addition, the meet-in-the-middle attack also manages to reduce the search time using $\#\mathcal{L}_f$ and $\#\mathcal{L}_\phi$ from the brute force attack, as follows.

$$\binom{\text{Key}}{\text{Security}} = \sqrt{\#\mathcal{L}_f} = \frac{1}{(d_f - 1)!}\sqrt{\frac{N!}{(d_f)(N - 2d_f + 1)!}}, \tag{20}$$

$$\binom{\text{Message}}{\text{Security}} = \sqrt{\#\mathcal{L}_\phi} = \frac{1}{d!}\sqrt{\frac{N!}{(N - 2d)!}}. \tag{21}$$

Note that the search time is reduced by squaring root $\#\mathcal{L}_f$ and $\#\mathcal{L}_\phi$ to obtain the key and message security for the proposed scheme. Thus, the following table will give the security level of the keys and message for the proposed scheme using the number of elements in the search space for the private key and the message in Table 3.

Table 4
The Key and Message Security for NTRU and the Proposed Scheme.

Level of Security	NTRU (Mittal, Kumar, Narain, & Kumar, 2021)		Proposed scheme	
	$\sqrt{\#\mathcal{L}_g}$	$\sqrt{\#\mathcal{L}_\phi}$	$\sqrt{\#\mathcal{L}_f}$	$\sqrt{\#\mathcal{L}_\phi}$
Moderate	1.2×10^{15}	9.4×10^{7}	9.7×10^{16}	9.4×10^{7}
High	9.2×10^{24}	2.1×10^{23}	2.5×10^{38}	2.1×10^{23}
Highest	6.1×10^{85}	4.8×10^{72}	2.3×10^{108}	4.8×10^{72}

4.4 Multiple Transmission Attacks

Multiple transmission attacks were first presented in (Jeffrey Hoffstein et al., 1998) and then analyzed further in (Jeffery Hoffstein, 1996). This attack could recover the message when a message has been sent multiple times using the same public key with different random polynomial for each time.

Theorem 5 *The proposed scheme is secure against multiple transmission attacks.*

Proof. Let the encrypted message be $e_i(X) = \big(p\phi_i(X) + m(X)\big) * h(X) \pmod q$ for $1 \le i \le N - 2$. Assume that the messages were encrypted several times by a single public key h and different random polynomial ϕ_i where ϕ_i be a sequence of random choices of polynomials in $\mathcal{L}(d, d)$. Then, the attacker will compute the following.

$e_i(X) - e_1(X) \equiv (p\phi_i(X) + m(X)) * h(X) - (p\phi_1(X) + m(X)) * h(X) \pmod q \equiv p(\phi_i(X) - \phi_1(X)) * h(X) \pmod q$. Rearranging the latter inequality yields $\phi_i(X) - \phi_1(X) = \frac{1}{p}\big(e_i(X) - e_1(X)\big) * h^{-1}(X) \pmod q$.

Note that the coefficients of the ϕ's are ranging from -1 to 1. To get the possibility for the coefficient of $\phi_i - \phi_1$, let , α be the *k*th coefficient of ϕ_i, β be the *k*th coefficient of ϕ_1 and α - β be *k*th coefficient of $\phi_i - \phi_1$. Then those possibilities are listed in the following table.

Table 5
The Coefficient of $\boldsymbol{\phi_i - \phi_1}$

	β		
α	-1	0	1
-1	0	-1	-2
0	1	0	-1
1	2	1	0

Table 5 shows that an attacker manages to recover approximately 1/9 of the coefficients of $\phi_i - \phi_1$, that is α – β = 2, by deducing β = -1. Similarly, an attacker manages to recover approximately 1/9 of coefficients of $\phi_i - \phi_1$, that is, α – β = -2 by deducing β = 1. Next, an attacker manages to recover approximately 2/9 of coefficients of $\phi_i - \phi_1$, that is, α – β = 1 by deducing β = -1,0. Similarly, an attacker manages to recover approximately 2/9 of coefficients of $\phi_i - \phi_1$, that is, α – β = -1 by deducing β = 0,1. This shows that an attacker should be able to recover almost every coefficient of $\phi_i - \phi_1$, and obtain exactly the values of $\phi_i - \phi_1$. Hence, an attacker can recover the single message *m* corresponding to ϕ_1 by using the brute force attack. However, this single message *m* corresponding to ϕ_1 is only a small part of the message and does not contain an information for any subsequent messages. This implies that an attacker would only manage to decrypt a single part of message *m* and not able to decrypt the whole message *m*. Therefore, the proposed scheme is secure against multiple transmission attacks.

4.5 Lattice Attack

In the lattice attack, an attacker will attack the NTRU cryptosystem through the public key h using LLL algorithm to recover the private keys f and g. Recall that the NTRU public key h is generated using the private keys f and g by the relationship $f(X) * h(X) = g(X) \pmod q$. In addition, the NTRU public key h associated by the NTRU lattice, $\mathcal{L}_h^{NTRU}$ can derive a basis of $\mathcal{L}_h^{NTRU}$, which represent the private keys f and g. For this lattice attack, the proposed scheme used a similar argument to define the proposed lattice $\mathcal{L}_{\hbar}^{proposed}$ associated to a public key $\hbar$, which satisfies $(X) \equiv f_q^{-1}(X) * r(X) * s_q^{-1}(X) \pmod q$ where f, r and s are private keys.

Definition 4 *The proposed lattice, $\mathcal{L}_{\hbar}^{proposed}$ associated to a proposed public key $\hbar(X) = \hbar_0 + \hbar_1 X + \hbar_2 X^2 + \cdots + \hbar_{N-1} X^{N-1}$ is the $2N$-dimensional lattice generated by the rows of the following block matrix*

$$\mathcal{L}_{\hbar}^{proposed} = \left(\begin{array}{cccc|ccccc} 1 & 0 & \cdots & 0 & \hbar_0 & \hbar_1 & \cdots & \hbar_{N-1} \\ 0 & 1 & \cdots & 0 & \hbar_{N-1} & \hbar_0 & \cdots & \hbar_{N-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & \hbar_1 & \hbar_2 & \cdots & \hbar_0 \\ \hline 0 & 0 & \cdots & 0 & q & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & q \end{array}\right),$$

which composed by four N–by-N quadrants:
Upper left quadrant = Identity matrix,
Upper right quadrant = Circulant matrix $\hbar$ (Cyclic permutation of the coefficients of $\hbar(X)$)
Lower left quadrant = Zero matrix,
Lower right quadrant = q times the identity matrix.

Note that the proposed lattice $\mathcal{L}_{\hbar}^{proposed}$ is convenient to abbreviate as a 2-by-2 block matrix with coefficients in *R* as follows: $\mathcal{L}_{\hbar}^{proposed} = \left(\frac{I}{0} \middle| \frac{\hbar}{qI}\right) \equiv M_{\hbar}^{proposed}$, which also called the proposed matrix $M_{\hbar}^{proposed}$.

The above definition will be used in the following result.

Proposition 5 *Suppose that $f(X) * s(X) * \hbar(X) \equiv r(X) \pmod q$. For any $u(X) \in R$, let $f(X) * s(X) * \hbar(X) = r(X) + q * u(X)$, then, $(f * s \quad -u) M_{\hbar}^{proposed} = (f * s \quad r)$. Therefore, the vector $(f * s \quad r)$ is in $\mathcal{L}_{\hbar}^{proposed}$.*

Proof. Let choose $(f * s \quad -u)$ be a row vector. The multiplication of a vector $(f * s \quad -u)$ with a proposed matrix $M_{\hbar}^{proposed}$ yields $(f * s \quad -u)\left(\frac{I}{0} \middle| \frac{\hbar}{qI}\right) = (f * s \quad f * s * \hbar - q * u)$. In particular, when the row vector $(f * s \quad -u)$ is multiply with the first column of proposed matrix $M_{\hbar}^{proposed}$, which consists of the identity matrix at the first quadrant followed by the zero matrix at the second quadrant, produces the vector $f * s$.

Next, multiply the vector $(f * s \quad -u)$ with the second column of proposed matrix $M_{\hbar}^{proposed}$, which is the circular matrix of $\hbar$ at the first quadrant followed by the identity matrix multiply with q at the second quadrant, yields the vector $(f * s \quad f * s * \hbar - q * u)$. Note that the kth entry of the vector $(f * s * \hbar - q * u)$ can be represented as $f_0 * s_0 * \hbar_k + f_1 * s_1 * \hbar_{k-1} + \cdots + f_{N-1} * s_{N-1} * \hbar_{k+1} - q * u_k$. From (4), the kth entry of the vector $(f * s * \hbar - q * u)$ is equal to the kth entry of the vector r. Thus, $(f * s \quad f * s * \hbar - q * u) = (f * s \quad r)$ since the vector $(f * s \quad r)$ is a linear combination of the column of the matrix $M_{\hbar}^{proposed}$, then the vector $(f * s \quad r)$ is in $\mathcal{L}_{\hbar}^{proposed}$.

From Proposition 5, it is clear that the vector $(f * s \quad r)$ is in $\mathcal{L}_{\hbar}^{proposed}$. It is also know that the coefficient of the vector $(f * s \quad r)$ is small, then the vector $(f * s \quad r)$ can be a short nonzero vector in $\mathcal{L}_{\hbar}^{proposed}$ or can be the shortest nonzero vector in $\mathcal{L}_{\hbar}^{proposed}$.

Proposition 6 *Let* (N, p, q, d_f, d_r, d_s) *be proposed parameters, with assumption* $p = 3, \quad d_f = d_r = d_s \approx \frac{N}{3}, \quad and \quad q \approx 2pN$.

Let $\mathcal{L}_{\hbar}^{proposed}$ *be a proposed lattice associated to the vector* $(f * s \quad r)$. *Then we have the following*

(a) $\det(\mathcal{L}_{\hbar}^{proposed}) = q^N$.

(b) $\|(f * s \quad r)\| \approx 0.471\sqrt{N(2N+3)}$.

(c) The Gaussian heuristic predicts that the expected shortest nonzero vector length of $\mathcal{L}_{\hbar}^{proposed}$ *is* $\sigma(\mathcal{L}_{\hbar}^{proposed}) = \sqrt{\frac{6}{\pi e}}N$. *If* N *is large, then there is a high probability that the shortest nonzero vector in* $\mathcal{L}_{\hbar}^{proposed}$ *are* $(f * s \quad r)$ *and its rotations. Further,* $\frac{\|[f*s \quad r]\|}{\sigma(\mathcal{L}_{\hbar}^{proposed})} \approx 0.562\sqrt{2 + \frac{3}{N}}$ *,*
so, the vector $(f * s \quad r)$ *is a factor of* $O(\sqrt{2 + 3/\mathrm{N}})$ *shorter than predicted by the Gaussian heuristic.*

Proof.

(a) Since the proposed matrix $M_{\hbar}^{proposed}$ is an upper triangular block matrix, then its determinant is the product of the diagonal entries. Its diagonal entries consist of N blocks of the identity matrix and N blocks of the identity matrix multiply with q. Therefore, $\det(\mathcal{L}_{\hbar}^{proposed}) = 1^N \times q^N = q^N$.

(b) It is known that $f(X) \in \mathcal{L}(d_f, d_f - 1)$, $r(X) \in \mathcal{L}(d_r, d_r + 1)$ and $s(X) \in \mathcal{L}(d_s + 1, d_s)$.Then, $\|(f * s \quad r)\| = \sqrt{\|f * s\|^2 + \|r\|^2} = \sqrt{(2d_f - 1)(2d_s + 1) + (2d_r + 1)}$Using $d_f = d_r = d_s \approx \mathrm{N}/3$ in the above inequality yields $\|(f * s \quad r)\| = \sqrt{\frac{2}{9}(2N^2 + 3N)}, = 0.471\sqrt{N(2N+3)}$.

(c) Since the dimension of $\mathcal{L}_{\hbar}^{proposed}$ is $2N$, then apply this and the condition (a) in Definition 1 will give the Gaussian expected shortest length of $\mathcal{L}_{\hbar}^{proposed}$ as follows.

$$\sigma(\mathcal{L}_{\hbar}^{proposed}) = \sqrt{\frac{2N}{2\pi e}}(q^N)^{1/2N} = \sqrt{\frac{Nq}{\pi e}}. \tag{22}$$

Thus, the above equation becomes Eq. (5) when $q = 6N$ where $p = 3$. Note that $\sigma(\mathcal{L}_{\hbar}^{proposed})$ will produces a large value when N is large. By Proposition 5, a vector $(f * s \quad r)$ can be a short vector in $\mathcal{L}_{\hbar}^{proposed}$. Thus, the vector $(f * s \quad r)$ and its rotations will have high possibility to be the shortest nonzero vectors in $\mathcal{L}_{\hbar}^{proposed}$ when N is large.

Generally, the Gaussian heuristic predicts that $\|v_{shortest}\| \approx \sigma(\mathcal{L})$ and the ratio of $\|v_{shortest}\|$ to $\sigma(\mathcal{L})$ is equal to 1. Using the fact that the vector $(f * s \quad r)$ is the shortest nonzero vectors in $\mathcal{L}_{\hbar}^{proposed}$ when N is large, Gaussian heuristic predicts that $\|(f * s \quad r)\| \approx \sigma(\mathcal{L}_{\hbar}^{proposed})$ and the ratio of $\|(f * s \quad r)\|$ to $\sigma(\mathcal{L}_{\hbar}^{proposed})$, which using the condition (b) and eq. (5) yields $\frac{\|(f*s \quad r)\|}{\sigma(\mathcal{L}_{\hbar}^{proposed})} \approx \frac{0.471\sqrt{N(2N+3)}}{\sqrt{\frac{6}{\pi e}}N} \approx 0.562\sqrt{2 + \frac{3}{N}}$.

As N grows large, an upper bound on the ratio of $\|(f * s \quad r)\|$ to $\sigma(\mathcal{L}_{\hbar}^{proposed})$ is $O(\sqrt{2 + 3/\mathrm{N}})$. Therefore, the vector $(f * s \quad r)$ is a factor of $O(\sqrt{2 + 3/\mathrm{N}})$ shorter than that predicted by the Gaussian heuristic.

Theorem 6 *The proposed scheme is secure against lattice attack.*

Proof. Proposition 6 tell us that the vector $(f * s \quad r)$ and its rotations will have high possibility to be the shortest nonzero vectors in $\mathcal{L}_{\hbar}^{proposed}$ when N is large. Since the private key $(f * s \quad r)$ is generating the public key $\hbar$, then the vector $\hbar$ also has high probability to be the shortest nonzero vector in $\mathcal{L}_{\hbar}^{proposed}$. Then there is a possibility for the LLL algorithm to attack the public key $\hbar$ with the first row of the LLL reduced basis corresponds to the vector $(f * s \quad r)$. Assume that LLL algorithm can attack the public key $\hbar$ in $\mathcal{L}_{\hbar}^{proposed}$ if the vector $(f * s \quad r)$ becomes the shortest nonzero vector in $\mathcal{L}_{\hbar}^{proposed}$. By Theorem 1 (Hermite's theorem), the length of a shortest nonzero vector $(f * s \quad r)$ in $\mathcal{L}_{\hbar}^{proposed}$ satisfies $\|(f * s \quad r)\| \leq \sqrt{2N}(q^N)^{\frac{1}{2N}} = \sqrt{2Nq}$.

This explicit bound is assumed as the LLL reduce basis bound. As N and q grows larger, then LLL algorithm have a difficulty to produce LLL reduced basis. Thus, LLL algorithm hardly recover the vector $(f * s \quad r)$ and consequently, LLL algorithm fails to attack the public key $\hbar$. This contradicts the assumption that the LLL algorithm able attack the vector $\hbar$ in $\mathcal{L}_{\hbar}^{proposed}$ when the vector $(f * s \quad r)$ becomes the shortest nonzero vector in $\mathcal{L}_{\hbar}^{proposed}$. Therefore, the proposed scheme is secure against lattice attack.

Numerical Illustration of Lattice Attack on Proposed Scheme

Consider the parameter $(N, p, q) = (7,3,47)$ and the public key $\hbar(X) \equiv 7X^6 + 18X^5 + 10X^4 + 19X^3 + 44X^2 + 18X + 24 \pmod{47}$. Performing LLL algorithm on the public key $\hbar$ yields a reduced basis as follows.

$$\mathcal{L}' = \begin{pmatrix} 0 & 1 & 0 & -1 & -1 & 1 & -1 & 0 & 1 & 1 & 0 & 0 & -1 & 0 \\ -1 & 0 & 1 & 0 & -1 & -1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & -1 \\ -1 & -1 & 1 & -1 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 1 & 0 & -1 & -1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & -1 & -1 & 1 & -1 & 0 & 1 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -1 & 1 & -1 & 0 & 1 & 1 & 0 & 0 & -1 & 0 & 0 & 1 \\ 1 & -1 & 1 & 0 & -1 & 0 & 1 & 0 & 1 & 0 & 0 & -1 & -1 & 0 \\ 5 & -4 & 4 & 2 & 7 & 7 & 2 & 9 & -3 & 9 & 6 & 3 & -2 & 1 \\ -6 & -2 & -5 & 0 & 5 & 4 & 5 & -2 & 10 & -5 & 0 & 10 & -6 & -8 \\ 5 & 4 & 5 & -6 & -2 & -5 & 0 & 10 & -6 & -8 & -2 & 10 & -5 & 0 \\ 2 & 7 & 7 & 2 & 5 & -4 & 4 & 6 & 3 & -2 & 2 & 9 & -3 & 9 \\ -7 & -7 & -2 & -5 & 4 & -4 & -2 & -3 & 2 & -2 & -9 & 3 & -9 & -6 \\ 3 & 2 & 8 & 7 & 1 & 4 & -3 & 9 & 6 & 4 & -1 & 2 & 9 & -4 \\ -7 & -2 & -5 & 4 & -4 & -2 & -7 & 2 & -2 & -9 & 3 & -9 & -6 & -3 \end{pmatrix}$$

The first row of LLL reduced basis represent

$$(f * s \quad r) = (0 \quad 1 \quad 1 \quad -1 \quad -1 \quad 1 \quad -1 | 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad -1 \quad 0)$$

which corresponding to $f(X) * s(X) = X - X^3 - X^4 + X^5 - X^6$, and $r(X) = X + X^2 - X^5$.

Note that the obtained LLL reduced basis will represent the vector $(f * s \quad r)$ when $f(X) * s(X) * \hbar(X) = r(X)$. To verify this, let compute $f(X) * s(X) * \hbar(X) \equiv 43X^6 + 12X^5 + 13X^4 + 21X^3 + 31X^2 + 18X + 4 \pmod{47} \neq -X^5 + X^2 + X = r(X)$. This shows that the obtained LLL reduced basis is not able to represent the vector $(f * s \quad r)$. Therefore, LLL algorithm fails to attack the public key $\hbar$.

5. Comparison between the proposed scheme and the NTRU

This section will give a comparison in terms of the computational complexity and the security level of the proposed scheme with the NTRU cryptosystem.

5.1 Computational Complexity

The following table shows the computational complexity based on the arithmetic operations (polynomial addition and convolution multiplication) on proposed scheme and the NTRU cryptosystem.

Table 6
The Arithmetic Operation of the Proposed Scheme and the NTRU.

	Proposed scheme	NTRU
Key Generation	2 convolution multiplication	1 convolution multiplication
Encryption	2 convolution multiplication and 1 polynomials addition	1 convolution multiplication and 1 polynomials addition
Decryption	2 convolution multiplication and 1 polynomials addition	2 convolution multiplication and 1 polynomials addition

Based on Table 6, let t be the time of convolution multiplication and t_1 be the time of polynomial addition, then the speed of the proposed scheme and the NTRU cryptosystem is obtained as follows.

Table 7
Speed of the Proposed Scheme and the NTRU

	Proposed scheme	NTRU
Speed	$6t + 2t_1$	$4t + 2t_1$
	$O(N^2)$ operations	$O(N^2)$ operations

In terms of speed, the proposed scheme is a little bit slow compared to the NTRU cryptosystem. But the proposed scheme as efficient as the NTRU cryptosystem.

5.2 Security Level

The following table shows that the security level of the private key and message for the proposed scheme and the NTRU cryptosystem.

Table 8
The Key and Message Security of the Proposed Scheme and the NTRU

	Proposed scheme	NTRU
Key security	$\frac{1}{(d_f-1)!}\sqrt{\frac{N!}{(d_f)(N-2d_f+1)!}}$	$\frac{1}{d_g!}\sqrt{\frac{N!}{(N-2d_g)!}}$
Message security	$\frac{1}{d!}\sqrt{\frac{N!}{(N-2d)!}}$	$\frac{1}{d!}\sqrt{\frac{N!}{(N-2d)!}}$

For the same coefficients, the key security level of the proposed scheme is a litter bit higher than the NTRU cryptosystem. But the message security of the proposed scheme is as the same as the NTRU cryptosystem.

6. Conclusion

In this work, a new NTRU variant is proposed, and some security analyses were performed to ensure the proposed variant is at least as secure as in the original NTRU cryptosystem. According to the analyses done in this work, it turns out that the proposed NTRU variant has a higher key security level as compared to the original NTRU cryptosystem, and at same time maintaining the message security level. Along the way, we have established several properties related to the security aspects, against some known attacks namely, the alternate private keys attack, brute force attack, meet-in-the-middle attack, multiple transmission attacks and lattice attack. NTRU cryptosystem is known as a lattice-based cryptosystem and has attracted many researchers to continue working on it. Many NTRU variants have been developed, and implemented, since it was first initiated in 1996 and more are still underway including this proposed scheme. It is believed that the proposed scheme is secure against the known attacks and will be able to survive until the next attacks.

Funding

This research is funded by the Fundamental Research Grant Scheme (FRGS), FRGS/1/2020/STG06/USM/01/1 from the Ministry of Higher Education Malaysia.

References

Abouaroek, M., & Ahmad, K. (2021). Performance analysis of NTRU algorithm with non-post-quantum algorithms. *Journal of Discrete Mathematical Sciences and Cryptography, 24*(5), 1349-1363.

Abu-Ulbeh, W., Altalhi, M., Abualigah, L., Almazroi, A. A., Sumari, P., & Gandomi, A. H. (2021). Cyberstalking victimization model using criminological theory: A systematic literature review, taxonomies, applications, tools, and validations. *Electronics, 10*(14), 1670.

Ahmad, K., Kamal, A., Ahmad, K. A. B., Khari, M., & Crespo, R. G. (2021). Fast hybrid-MixNet for security and privacy using NTRU algorithm. *Journal of Information Security and Applications, 60*, 102872.

Ajtai, M., & Dwork, C. (1997). A public-key cryptosystem with worst-case/average-case equivalence. *Paper presented at the Proceedings of the twenty-ninth annual ACM symposium on Theory of computing.*

Al-Saidi, N. M., & Yassein, H. R. (2017). A new alternative to NTRU cryptosystem based on highly dimensional algebra with dense lattice structure. *Malaysian Journal of Mathematical Sciences, 11*, 29-43.

Alshurideh, M., & Kurdi, B. (2023). Factors affecting social networks acceptance: An extension to the technology acceptance model using PLS-SEM and Machine Learning Approach. *International Journal of Data and Network Science, 7*(1), 489-494.

Barsocchi, P., Calabrò, A., Crivello, A., Daoudagh, S., Furfari, F., Girolami, M., & Marchetti, E. (2021). COVID-19 & privacy: Enhancing of indoor localization architectures towards effective social distancing. *Array, 9*, 100051.

Gaubatz, G., Kaps, J.-P., & Sunar, B. (2004). Public key cryptography in sensor networks—revisited. *Paper presented at the European Workshop on Security in Ad-Hoc and Sensor Networks*.

Goldreich, O., Goldwasser, S., & Halevi, S. (1997). Public-key cryptosystems from lattice reduction problems. *Paper presented at the Annual International Cryptology Conference.*

Hoffstein, J. (1996). NTRU: a new high speed public key cryptosystem. presented at the rump session of Crypto 96.

Hoffstein, J., Pipher, J., & Silverman, J. H. (1998). *International Algorithmic Number Theory Symposium*. In: Springer: Berlin, Germany.

Hoffstein, J., Pipher, J., Silverman, J. H., & Silverman, J. H. (2008). *An introduction to mathematical cryptography* (Vol. 1): Springer.

Howgrave-Graham, N. (2007). A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. *Paper presented at the Annual International Cryptology Conference*.

Huang, D., Zhou, J., Mi, B., Kuang, F., & Liu, Y. (2022). Key-based data deduplication via homomorphic NTRU for internet of vehicles. *IEEE Transactions on Vehicular Technology.*

Imam, R., Areeb, Q. M., Alturki, A., & Anwer, F. (2021). Systematic and critical review of rsa based public key cryptographic schemes: Past and present status. *IEEE Access*.

Jarah, B., Jarrah, M., Almomani, S., AlJarrah, E., & Al-Rashdan, M. (2023). The effect of reliable data transfer and efficient computer network features in Jordanian banks accounting information systems performance based on hardware and software, database and number of hosts. *International Journal of Data and Network Science, 7*(1), 357-362.

Kaaniche, N., Laurent, M., & Belguith, S. (2020). Privacy enhancing technologies for solving the privacy-personalization paradox: Taxonomy and survey. *Journal of Network and Computer Applications, 171*, 102807.

Kalra, S., & Sood, S. K. (2015). Secure authentication scheme for IoT and cloud servers. *Pervasive and Mobile Computing, 24*, 210-223.

Mittal, G., Kumar, S., & Kumar, S. (2021). Novel public-key cryptosystems based on NTRU and algebraic structure of group rings. *Journal of Information and Optimization Sciences, 42*(7), 1507-1521.

Mittal, G., Kumar, S., Narain, S., & Kumar, S. (2022). Group ring based public key cryptosystems. *Journal of discrete mathematical sciences and cryptography*, *25*(6), 1683-1704.

Mughaid, A., Al-Zu'bi, S., Al Arjan, A., Al-Amrat, R., Alajmi, R., Zitar, R. A., & Abualigah, L. (2022). An intelligent cybersecurity system for detecting fake news in social media websites. *Soft Computing, 26*(12), 5577-5591.

Otair, M., Ibrahim, O. T., Abualigah, L., Altalhi, M., & Sumari, P. (2022). An enhanced grey wolf optimizer based particle swarm optimizer for intrusion detection system in wireless sensor networks. *Wireless Networks, 28*(2), 721-744.

Prasadh, K., Ramar, K., & Gnanajeyaraman, R. (2009). Public key cryptosystems based on chaotic-chebyshev polynomials. *Paper presented at the 2009 International Conference on Intelligent Agent & Multi-Agent Systems.*

Qin, X., Huang, R., & Fan, H. (2021). An effective NTRU-based fully homomorphic encryption scheme. *Mathematical Problems in Engineering*, 2021.

Regev, O. (2009). On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM), 56*(6), 1-40.

Salman, H. S., & Yassein, H. R. (2022). An Innovative HSS Algebra for Designing a Secure Like-NTRU Encryption. *Mathematical Statistician and Engineering Applications, 71*(4), 6098-6113.

Sever, M., & Özdemir, A. Ş. (2021). A new offer of NTRU cryptosystem with two new key pairs. *Numerical Methods for Partial Differential Equations, 37*(2), 1222-1233.

van Vredendaal, C. (2016). Reduced memory meet-in-the-middle attack against the NTRU private key. *LMS Journal of Computation and Mathematics, 19*(A), 43-57.

Xu, G., Dong, J., Ma, C., Liu, J., & Cliff, U. G. O. (2022). A Certificateless Signcryption Mechanism Based on Blockchain for Edge Computing. *IEEE Internet of Things Journal*.

Yassein, H. R., Al-Saidi, N. M., & Farhan, A. K. (2022). A new NTRU cryptosystem outperforms three highly secured NTRU-analog systems through an innovational algebraic structure. *Journal of Discrete Mathematical Sciences and Cryptography, 25*(2), 523-542.