National College of Ireland

QSTRU: A new variation of the NTRU public key cryptosystem

Academic Internship
MSc Cybersecurity

Success Jimoh
Student ID: X20139471

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……Success Jimoh……

Student ID: …… ……. 20139471……

Programme: ……MSc. Cybersecurity…… **Year:** ……2021……

Module: ……Academic Internship……

Supervisor: ……Vikas Sahni ……

Submission Due Date: ……16/12/2021……

Project Title: QSTRU: A new variation of the NTRU public key cryptosystem……

Word Count: ……6479…… **Page Count**……21……

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……Success Jimoh……

Date: ……16/12/2021……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

QSTRU: A new variation of the NTRU public key cryptosystem

Jimoh Daodu Success
20139471

Abstract

The invention of quantum computers will lead to current classical encryption techniques such as RSA, SHA, and DES being rendered inefficient against such quantum systems. Therefore, the search for a secure quantum algorithm that would be able to withstand attacks from these quantum systems is an active area of research these days. NTRU is currently the leading algorithm in the post-quantum cryptography space, and various variations of this algorithm have been created to improve upon it. In this paper a variation of the NTRU algorithm using a complex algebra called trigintaduonions is presented. This approach is shown to be more secure against attacks as opposed to NTRU and STRU.

Keywords: NTRU, STRU, trigintaduonions, complex algebra, post-quantum cryptography, lattice-based cryptography

1. Introduction

The development of quantum computing threatens to break many conventional encryption algorithms. This leads to public-key cryptographic developments that focus on basic post-quantum cryptography and quantum computing resistant protocols. For online interaction, automobiles, and IoT devices, cryptography is important. However, when quantum computers become commercially available, many currently used algorithms will get broken easily. This is based on the fact that current encryption algorithms work on the computational capacity of current classical computers, which makes it nearly impossible to break them or requires a long time to break. With the likely computational speed of quantum computing, the computational hardness of the current cryptographic algorithms can be broken easily. In this relatively recent field of study in post-quantum cryptography, mathematical processes were identified, for which quantum algorithms give no speed benefit and cryptographic systems were subsequently created.

Various cryptography schemes and algorithms have been suggested to be used against quantum computer attacks, but with the limitation of their being, this can't be proven. There have been multiple approaches of computer cryptography that have been suggested such as multivariate cryptography, code-based cryptography, has-based cryptography, and lattice-based cryptography

Lattice-based cryptography is an area of interest for post-quantum computing (PQC) due to the fact of the computational hardness problem it provides. One of the leading algorithms is NTRU “nth-polynomial ring”.

This research paper presents a new variation of the NTRU, where the nth-polynomial rings of NTRU are replaced with a new mathematical structure algebra called, trigintaduonions algebra. This is proposed to be an improved version of the STRU (Thakur & Tripathi, 2017). Trigintaduonions algebra was used for our cryptographic algorithm and compare the efficiency (time complexity) and security (space complexity) with STRU and NTRU cryptography algorithms. This paper determines the effectiveness of a variation of NTRU with a complex algebra of the non-associative finite invertible loop against the standard NTRU and STRU.

This paper answers the following research questions;
1) How would a higher degree of complex algebra used in a variation of the NTRU improve the cryptosystem?
2) How efficient will the trigintaduonions algebra be compared against the nth-polynomial algorithm?

This study includes various parts such as a literature review discussing past research in the areas of lattice-based cryptography, a methodology outlining the application of this research, design specification with details on the programming languages, libraries, and a description of the algorithm. It will also include a section for model implementation, which discussed the implementation of the proposed algorithm; an evaluation on metrics used for the algorithm and the comparison with STRU and NTRU; and lastly the outcomes of the research and other areas that relate and can be further researched.

2. Literature Review

Several researchers suggested various approaches that could be used to protect information against the future of computers from quantum computing attacks,. These approaches include lattice-based cryptography, multivariate cryptography, hash-based cryptography, and code-based cryptography (Buchmann et al., 2016). It is useful to first look at key subjects such as quantum computing, other approaches to post-quantum cryptography, and lattice cryptography, before assessing the research state of lattice-based algorithms.

2.1 Quantum Computing

Quantum computing is a field of the computer that is based on quantum theory concepts that advance computing technologies. Quantum computers are devices that perform quantum computations. For these computers, it is believed that they would solve computational problems faster and in polynomial time, rather than the current classical computers. Such computational problems that are expected to be solved by this quantum computer includes problems such as integer factorization, discrete log problem, lattice problems, and other computational problems. (Rieffel & Polak, 2000)

There are several forms of quantum computers, including a quantum circuit model, a quantum turing system, adiabatic quantum computer, and a series of quantum cellular automata. The most common model employed is the quantum circuit, which is based on the quantum bit or "qubit". Classical computers are somewhat comparable to the quantum bit. A qubit might be in a quantum state of 1 or 0 or a 1 and 0 state superposition. However, the

likelihood of either result will depend on the quantum states of the qubit shortly before measurement.

Significant use of quantum calculation is used for attacks on in-use encryption systems. Computationally impossible factorization, which underlies the safety of the publicly available key cryptography systems, with a normal computer for big integer systems, are considered to be the product of few prime numbers. By using Shor's algorithm (Shor, 1994), a quantum computer might effectively resolve this problem. This would allow a quantum computer to break down many of the cryptographic methods used today that have a polynomial-time algorithm to solve the problem.

2.2 Multivariate Cryptography

The generic term for asymmetric primitive cryptography is multivariate cryptography and is based on a finite field. These polynomials in certain circumstances can be defined both overground and over an extension field. These are multivariate quadratics if the polynomials are by the degree two. The systems that tend to be derived from multivariate polynomial equations are said to be nondeterministic polynomial-time complete. (Ding & Petzoldt, 2017) This includes cryptographic methods like Rainbow (Unbalanced Oil and Vinegar) based on multivariate equation solutions. Several efforts have failed to develop safe multivariate encryption equation methods. However, the cornerstone for a quantitatively safe digital signature may be multivariate signature systems such as Rainbow. For efficiency reasons, a system of quadratic polynomials over a small, finite K field with q elements are generally a public key of the multivariate public key cryptography

$$p^{(1)}(x_1 \ldots, x_n) = \sum_{i=1}^{n} p_{ij}^{(1)} x_i \cdot x_j + \sum_{i=1}^{n} p_i^{(1)} x_i + p_0^{(1)},$$
$$p^{(2)}(x_1 \ldots, x_n) = \sum_{i=1}^{n} p_{ij}^{(2)} x_i \cdot x_j + \sum_{i=1}^{n} p_i^{(2)} x_i + p_0^{(2)},$$
$$\ldots,$$
$$p^{(m)}(x_1 \ldots, x_n) = \sum_{i=1}^{n} p_{ij}^{(m)} x_i \cdot x_j + \sum_{i=1}^{n} p_i^{(m)} x_i + p_0^{(m)}.$$

Figure 1: Multivariate cryptography

2.3 Hash-based Cryptography

This cryptography approach was first proposed in 1987 by Ralph Merkle (Ralph, 1987) and it has been advancing ever since. This theory demonstrated that multiple one-time key pairs are merged with a hash tree into one structure. A hash tree is a hierarchical data structure that utilizes a hash and concatenation algorithm repeatedly to calculate nodes.

The reason why hash-based signature has been supported to be in for future of post quantum cryptography “PQC” is based on two-argument points ;

1) The security assumptions are very minimal: For schemes such as the Extended Merkle Signature Scheme (XMSS), it only takes a secure cryptographic hash function to ensure security. It is demonstrated that XMSS is secure as long as a secure signature algorithm

exists. In this respect, there are low-security requirements for XSMS and other hashed signature systems.
2) It has a generic nature: It is possible to instantiate hash-based signature schemes from any hash function that meets basic requirements, which provides huge flexibility. The hash functions underlying these schemes can be used to replace them without changing their basic structure. This tends to be important for security in the long run as various vulneruablities may emerge.
The cornerstone of hash-based signatures is one-time signature methods. It may be regarded as digital thumbnails that can only be utilized once for a certain key pair. Their safety is based only on the safeguards of the basic hash function. While this underlying hash function presupposes certain technological assumptions, many hash functions meet these requirements and many others are to be created in the future. (Butin, 2017)

2.4 Code-based Cryptography
Code-based encryption covers all cryptosystems, symmetric or asymmetric, whose security depends largely or entirely on decoding hardiness, if selected with some certain structure or in a certain family, in a linear error correction code (for instance, quasi-cyclic codes, or Goppa codes) (Sendrier, 2011). This comprised cryptographic systems is based on error correction codes, such as techniques of encryption of McEliece and Niederreiter (McEliece, 1978), the corresponding scheme of Courtois, Finiasz, and Sendrier Signature (Courtois et al., 2001).

A very strong candidate for future quantum-resistant standards for public-key encryption, is the original McEliece encryption scheme that must be developed in the next decade. Its primary drawback however, is a rather high key size, which makes it less suited for some applications (for the long-term, quantum-resistant variable of 1 Mbyte in order).

2.5 Lattice-Based Cryptography
This paper is centered on the lattice-based cryptographic algorithm, which looks into lattice-based cryptography and various works carried on about it in the past cryptography.

Lattice-based Cryptography is a general word for cryptographic basic constructs including lattice for building itself. A lattice is a regular collection of n-dimensional points generated by all vectors, which rely linearly on the so-called basis of a set of linearly independent vector b. (Mohsen et al., 2017).

$$\mathcal{L}\ (b1, ...,bn) = \left\{ \sum_{I=1}^{n} x_i b_i : x_i \in \mathbb{Z} \right\}$$

Figure 2 : Mathematical definition of a lattice

For the case of classical-based algorithms, it works with the average-case problems, while lattice-based works on the worst-case problems. One of the worst-case problems that lattice-based cryptography relies on, is the shortest vector problem. The given vector space V of the shortest vector problem is in this space, V a shortest vector must be found given by the norm N. A norm is a function that assigns each vector in a vector space an absolute or positive length. Another cooptation problem would be the closest vector problem, given the basis of a lattice and a vector v, not in the lattice, find the closest vector to v that is in the lattice.

2.6 Related Works

The lattice-based cryptography began in 1997 using the proven-security Ajtai-Dwork cryptosystems. The cryptosystem is based on the problem of the hidden plane (HPP) (Ajtai & Dwork, 1997) . This cryptosystem operates when a private key is specified as a secret n-dimensional vector for a positive integer n. The secret vector is defined as a periodic collection of equidistant parallel hyperplanes in the n-dimensional space: the secret vector has an inverse length proportion to the distance between the subsequent hyperplanes. The public key is specified as several hyperplanes points. In n-dimensional terms, every piece of plaintext is encrypted. The ciphertext is a uniform, random point if the message bit is 0. The ciphertext is the center of an even subset of public key points if the message bit is 1. Decryption is made by a projection of the ciphertext on a privately key secret vector (scalar product with unit vector),1 if a hyperplane is close to the point, and 0 if not. (Mohsen et al., 2017) . This system was very inefficient in working as an algorithm as it was just a proof of concept.

O. Goldreich, S. Goldwasser and S. Halevi presented their GGH crypto-system in 1997 (Goldreich et al., 1997). Compared with the Ajtai-Dwork, this increased the space efficiency and did not give a guarantee of security. The private key is a solid foundation, while the public key is a terrible base. A good foundation contains base vectors, which are virtually perpendicular and as short as feasible. A tiny vector adding a grid is represented as a message (a point linearly dependent based on the lattice). Then the module from the wrong base is decreased. The point is then reduced. By decreasing the cipher-text vector modulo, the basic parallel tube from the right basis, the message is readily decoded with the private key. But deciphering the message on a faulty foundation is comparable to resolving the nearest CVP issue, which is a difficult one. The shortest number of linearly independent vectors is a solid foundation. There may be vectors far from the origin, yet the same lattice is described. This crypto-system has no assurance of safety and other assaults have been found till the crypto-system is unsafe in some dimensions.

In 2005, O Regev presented an issue of the LWE (Learning with Errors) decision (Regev, 2009), one of the major difficulties in grid-based encryption. It states that in certain couples in the form (a[i],b[i]), in which a[i] is a polynomial value and b[i] is a single value, all of which must be modulo q, whether the term b[i] is uniformly random or depends on the first component a[i] plus a minor noise value, b[i]=a(i)s+e[i] where s is a secret vector and e[i] is

normally a small error vector from the Gaussian distribution. The usual LWE issue involves the same couples and the aim is to identify the secret vector.

The error terms have to be minimal enough in comparison with the floor(q/4) to ensure that message decryption succeeds. In the unusual case that certain Gaussian error terms turn out to be similar to q, the error terms are sampled on the Gaussian distribution with an adequately low standard deviation.

By extending the LWE issue into Compact-LWE problem[in 2017, D. Liu et al (Liu et al., 2017) enhanced Regev's first public-key cryptosystem. In other words, the problem on which the cryptosystem's difficulty is predicated is now more common. For certain parameter values, the original LWE crypto-system of Regev is under lattice-based attack. A private key can be recovered by solving a limited number of dimensions of the nearest vector problem (CVP). The CVP is known to be difficult, but it must be intractable in many ways Therefore, lattice-based attacks should be taken into account in Regev's LWE cryptosystem, while setting values for parameters. By working with denser grids, the efficiency of space is also increased. The public key of the original LWE cryptosystem of Regev was made up of the collection of samples (a[i], y[i]), a[i] being a vector in Zq kernel, with modulo q components, and y[i] being an integer in zq. While the basis modulus b is lower in the Compact-LWE encryption scheme, the vectors a[i] (Basis Matric AT rows) are smaller ingredients. The Compact-LWE cryptosystem also removes the possible decryption failure of the LWE crypto-system of Regev. Nevertheless, the Compact-LWE cryptosystem is still almost as inefficient as Regev.

A cryptosystem based on the LWE problem over polynomial rings (Ring-LWE) was presented in 2010, with an impact by the NTRU cryptosystem and the Micciancio one-way ring-SIS function. The Ring-LWE crypto-system employs laths, which are suitable for a polynomial ring forming R=Z[x]/f(x), where Z[x] is a series of x polynomials with coefficients in Z and f (x) is any polynomial selected for modular reduction (x). Rq=Zq[x]/ (xn+1) is the most often used ring with a 2 power n. This is consistent with several polynomials with module q n coefficients and x to n-1 power. The set of n-dimensioned anti-cyclic grids is perfect in the Ring Rq in this situation.

In 2016, a different version of NTRU was proposed by Ali. Majeed named the CQTRU(Al-Saidi et al., 2016), where the public cryptosystems were based on commutative quaternions algebra. This is a four-dimensional cryptosystem that was based on the commutative quaternion ring as opposed to the polynomial ring being used in NTRU.

Another approach to a different version to improve the NTRU was also proposed using the hexadecimal and binary algebras in 2016 by Yassein H R and Al-Saidi. It was called the HXDTRU cryptosystems. The HXDTRU was 16 times faster than the NTRU for security with a large N. (Al-Saidi & Yassein, 2016). The binary-based algebra that was used as a variant for NTRU is called Bitru. (Bitru, 2016). It was differentiated from the other cryptosystem as well as NTRU because it was establishing two public keys in the cryptosystems.

In 2017, Khushboo Thakur and B.P. Tripathi introduced STRU, (Thakur & Tripathi, 2017) a non-alternative and multi-dimensional public-key cryptosystem based on the NTRU, employing sedenion algebra. In each encryption method, the STRU encrypts sixteen data vectors. STRU's underlying algebraic structure is a non-associative and non-alternative 16-dimensional algebra with a quadratic form, whose elements are built from real numbers R using iterations of the Cayley–Dickson Process.

In 2018, Yassein and Al-Saidi proposed BCTRU, a new approach to NTRU-like cryptosystem based on bi-cartesian algebra. BCTRU is a multi-dimensional NTRU-like public-key cryptosystem that was recently created. It is built on an algebraic structure called bi-cartesian algebra, which is both commutative and associative, rather than the traditional NTRU-polynomial ring. (Yassein & Al-Saidi, 2018).

Yassein et al. (Yassein et al., 2020) developed a new NTRU cryptosystem called QOBTRU that uses multidimensional quaternion algebra. Based on this newly constructed algebraic structure Quaternion algebra, a new NTRU-analog cryptosystem QOBTRU is suggested. In terms of computational and spatial complexity, QOBTRU is at least quadratic higher than the original NTRU. QTRU, OTRU, and BITRU are three highly performing multidimensional NTRU-like cryptosystems that are meant to have alternate security and performance characteristics.

Yassein et al. (Yassein et al., 2021) introduced QMTRU in 2021, which is an enhancement on QTRU based on a new mathematical framework. A multi-dimensional QTRU public-key cryptosystem enhances NTRU by substituting quaternion algebra for the original ring in NTRU. QMNTR is a new mathematical structure that uses two public keys and five private keys to improve on QTRU. The public key system has been strengthened and made more secure as a result of this change.

This year 2021, a new version of NTRU was proposed using a noncommunicative variant ion NTRU called the QOTRU (Abo-Alsood & Yassein, 2021). It uses a new mathematical structure of two public keys and five private keys to work in the Qu-octonion subalgebra of octonion algebra. The public key system has become more secure and sophisticated as a result of this new structure.

2.7 Discussion

The research papers discussed above have introduced various approaches to lattice-based cryptography including NTRU, learning with errors (LWE), and also an improved version of this approach with the Ring-based LWE and other substituting mathematical algebraic structures for the different proposed versions of NTRU.

Algorithm	Features/ Advantages	Researchers	Year
Ajtai-Dwork cryptosystems	The cryptosystem is based on the problem of the hidden plane (HPP)	Ajtai-Dwork et al.	1997

Algorithm	Features/ Advantages	Researchers	Year
GGH cryptosytem	It increased the space efficiency of the NTRU and did not give a guarantee of security	O. Goldreich, et al .	1997
LWE	It was based of on the Learning with Errors	O Regev	2005
CQTRU	It made use of the commutative quaternions	Ali. Majeed	2016
HXDTRU	it was establishing two public keys in the cryptosystems.	Yassein H R and Al-Saidi	2016
STRU	a non-alternative and multi-dimensional public-key cryptosystem employing sedenion algebra	Khushboo Thakur and B.P. Tripathi	2017
Compact-LWE	It was based on the compact LWE problem over polynomial rings	D. Liu et al	2017
BCTRU	It was based on bi-cartesian algebra	Al-Saidi et al.	2018
QOBTRU	It was based on the multidimensional quaternion algebra.	Yassein et al.	2020
QMTRU	A multi-dimensional QTRU public-key cryptosystem with two public keys and 5 private keys	Yassein et al.	2021
QOTRU	It uses a new mathematical structure of two public keys and five private keys to work in the Qu-octonion subalgebra of octonion algebra	Abo-Alsood et al.	2021

Most of these works suggested a complex algebra approach to the NTRU, but were limited in the area that they never were implemented. This research paper, suggests an approach called QSTRU and would also make an implementation approach. QSTRU makes use of a different mathematical structure called trigintaduonions algebra that depends on the Cayley-Dickson of doubling the sedenion algebra to improve on the STRU public cryptosystems.

3. Research Methodology

For this proposed algorithm, the NTRU was looked into. The proposed complex algebra trigintaduonions algebra, which is the Algebra of Dimension 32 (trigintaduonions), is also used (Cawagas et al., 2009).

The creation of the NTRU public-key cryptosystem, which is high-speed, has a small key and requires little memory. It is based on polynomial multiplication and residues taking into account the relative processing capacity of the encoder and decoder, as well as the relative importance of speed vs the chance that an infrequent message may be undeciphered. In NTRU, the basic operations take place in the ring Z[x]/(xN 1), also known as the ring of convolution polynomials of rank N, where N is a prime number. Addition and multiplication in the ring of convolution polynomials need only O(N) and O(N2) operations, respectively. R = Z[x]/(xN 1), Rp = (Z/pZ)[x]/(xN 1), and Rq = (Z/qZ)[x]/(xN 1) are the three rings to define. A polynomial or its vector of coefficients can be used to express an element f of the rings R, Rp, and Rq: f = N1 fi.xi [f0, f1,..., fN1]. (Hoffstein et al., 1998).

For the key generation of this algorithm the set L and N,p,q are used for the public key and are known by everybody. The following two polynomials make up an NTRU public key/ private key pair:

Private key : F = $f(x) = \sum_{i=0}^{N-1} f_i \,.\, x^i = [f_1, f2.....f_n]$

$h(x) = \sum_{i=0}^{N-1} h_i \,.\, x^i = [h_1, h2.....h_n]$

Public key : $H = \sum_{i=0}^{k} f_i g_{k-1} + \sum_{i=k+1}^{N-1} f_i g_{N+k-1} = \sum_{i+j=k} f_i g_j$

To get this private key they would be chosen randomly from a set options in the set L.

The extended Euclidian procedure is used to find the inverse of f over Rp and Rq in the NTRU algorithm. Those two inverses are referred to as $f_p^{-1}$and f_q^{-1}, respectively.

While the private keys f, g, f_p^{-1}, f_q^{-1} are kept private, the public key H is gotten by performing a computation operation of :

H $= f_q^{-1} * g$

And the encryption is done by :

E = p.h $*$ φ + m(modq).

The decryption of the cipher text os gotten by

Step 1 : The receiving polynomial e is multiplied (convolved) by the private key f

Step 2 : a shifting polynomial is created

Step 3 : it is then multiplied by f_p^{-1}

The complex algebra used for this proposed algorithm was the trigintaduonions algebra as seen in (Cawagas et al., 2009). This algebra is being served as the complex 32D algebra which adds a more layer of security for the NTRU.

The Cayley-Dickson algebras C(complex numbers 2-D), H(quaternions 4-D), O(octonions 8-D), S(sedenions 16-D), and T(trigintaduonions 32-D)1 are realalgebras created using a doubling approach known as the Cayley-Dickson (C-D) process. As a result, the following C-D doubling chain: RcHcOcScT demonstrates that the trigintaduonions T include S,O,H,C, and R subalgebras. Sedenion is derived from the Latin word sexdecim, which means sixteen S represents the true sedenion or hexadecanions. The sedenion is a power-associative non-commutative, non-associative, non-alternative. Furthermore, because it has no divisors, it is neither a composition algebra nor a division algebra. The trigintaduonions T have zero divisors in all 16-dimensional subalgebras. Except for S, these intriguing algebraic structures have never been discovered before.

The addition of two sedenion is achieved by adding the respective coefficients (i.e., element-wise), whereas multiplication is determined by bilinearity and the base element's multiplication rule.
The multiplication table of the trigintaduonions algebra is given below ;

*	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	1	-0	3	-2	5	-4	-7	6	9	-8	-11	10	-13	12	15	-14	17	-16	-19	18	-21	20	23	-22	-25	24	27	-26	29	-28	-31	30
2	2	-3	-0	1	6	7	-4	-5	10	11	-8	-9	-14	-15	12	13	18	19	-16	-17	-22	-23	20	21	-26	-27	24	25	30	31	-28	-29
3	3	2	-1	-0	7	-6	5	-4	11	-10	9	-8	-15	14	-13	12	19	-18	17	-16	-23	22	-21	20	-27	26	-25	24	31	-30	29	-28
4	4	-5	-6	-7	-0	1	2	3	12	13	14	15	-8	-9	-10	-11	20	21	22	23	-16	-17	-18	-19	-28	-29	-30	-31	24	25	26	27
5	5	4	-7	6	-1	-0	-3	2	13	-12	15	-14	9	-8	11	-10	21	-20	23	-22	17	-16	19	-18	-29	28	-31	30	-25	24	-27	26
6	6	7	4	-5	-2	3	-0	-1	14	-15	-12	13	10	-11	-8	9	22	-23	-20	21	18	-19	-16	17	-30	31	28	-29	-26	27	24	-25
7	7	-6	5	4	-3	-2	1	-0	15	14	-13	-12	11	10	-9	-8	23	22	-21	-20	19	18	-17	-16	-31	-30	29	28	-27	-26	25	24
8	8	-9	-10	-11	-12	-13	-14	-15	-0	1	2	3	4	5	6	7	24	25	26	27	28	29	30	31	-16	-17	-18	-19	-20	-21	-22	-23
9	9	8	-11	10	-13	12	15	-14	-1	-0	-3	2	-5	4	7	-6	25	-24	27	-26	29	-28	-31	30	17	-16	19	-18	21	-20	-23	22
10	10	11	8	-9	-14	-15	12	13	-2	3	-0	-1	-6	-7	4	5	26	-27	-24	25	30	31	-28	-29	18	-15	-16	17	22	23	-20	-21
11	11	-10	9	8	-15	14	-13	12	-3	-2	1	-0	-7	6	-5	4	27	26	-25	-24	31	-30	29	-28	19	18	-17	-16	23	-22	21	-20
12	12	13	14	15	8	-9	-10	-11	-4	5	6	7	-0	-1	-2	-3	28	-29	-30	-31	-24	25	26	27	20	-21	-22	-23	-16	17	18	19
13	13	-12	15	-14	9	8	11	-10	-5	-4	7	-6	1	-0	3	-2	29	28	-31	30	-25	-24	-27	26	21	24	-23	22	-17	-16	-19	18
14	14	-15	-12	13	10	-11	8	9	-6	-7	-4	5	2	-3	-0	1	30	31	28	-29	-26	27	-24	-25	22	23	20	-21	-18	19	-16	-17
15	15	14	-13	-12	11	10	-9	8	-7	6	-5	-4	3	2	-1	-0	31	-30	29	28	-27	-26	25	-24	23	-22	21	20	-19	-18	17	-16
16	16	-17	-18	-19	-20	-21	-22	-23	-24	-25	-26	-27	-28	-29	-30	-31	-0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
17	17	16	-19	18	-21	20	23	-22	-25	24	27	-26	29	-28	-31	30	-1	-0	-3	2	-5	4	7	-6	-9	8	11	-10	13	-12	-15	14
18	18	19	16	-17	-22	-23	20	21	-26	-27	24	25	30	31	-28	-29	-2	3	-0	-1	-6	-7	4	5	-10	-11	8	9	14	15	-12	-13
19	19	-18	17	16	-23	22	-21	20	-27	26	-25	24	31	-30	29	-28	-3	-2	1	-0	-7	6	-5	4	-11	10	-9	8	15	-14	13	-12
20	20	21	22	23	16	-17	-18	-19	-28	-29	-30	-31	24	25	26	27	-4	5	6	7	-0	-1	-2	-3	-12	-13	-14	-15	8	9	10	11
21	21	-20	23	-22	17	16	19	-18	-29	28	-31	30	-25	24	-27	26	-5	-4	7	-6	1	-0	3	-2	-13	12	-15	14	-9	8	-11	10
22	22	-23	-20	21	18	-19	16	17	-30	31	28	-29	-26	27	24	-25	-6	-7	-4	5	2	-3	-0	1	-14	15	12	-13	-10	11	8	-9
23	23	22	-21	-20	19	18	-17	16	-31	-30	29	28	-27	-26	25	24	-7	6	-5	-4	3	2	-1	-0	-15	-14	13	12	-11	-10	9	8
24	24	25	26	27	28	29	30	31	16	-17	-18	-19	-20	-21	-22	-23	-8	9	10	11	12	13	14	15	-0	-1	-2	-3	-4	-5	-6	-7
25	25	-24	27	-26	29	-28	-31	30	17	16	19	-18	21	-20	-23	22	-9	-8	11	-10	13	-12	-15	14	1	-0	3	-2	5	-4	-7	6
26	26	-27	-24	25	30	31	-28	-29	18	-19	16	17	22	23	-20	-21	-10	-11	-8	9	14	15	-12	-13	2	-3	-0	1	6	7	-4	-5
27	27	26	-25	-24	31	-30	29	-28	19	18	-17	16	23	-22	21	-20	-11	10	-9	-8	15	-14	13	-12	3	2	-1	-0	7	-6	5	-4
28	28	-29	-30	-31	-24	25	26	27	20	-21	-22	-23	16	17	18	19	-12	-13	-14	-15	-8	9	10	11	4	-5	-6	-7	-0	1	2	3
29	29	28	-31	30	-25	-24	-27	26	21	20	-23	22	-17	16	-19	18	-13	12	-15	14	-9	-8	-11	10	5	4	-7	6	-1	-0	-3	2
30	30	31	28	-29	-26	27	-24	-25	22	23	20	-21	-18	19	16	-17	-14	15	12	-13	-10	11	-8	-9	6	7	4	-5	-2	3	-0	-1
31	31	-30	29	28	-27	-26	25	-24	23	-22	21	20	-19	-18	17	16	-15	-14	13	12	-11	-10	9	-8	7	-6	5	4	-3	-2	1	-0

Figure 3 : Multiplication table of the trigintaduonions (Cawagas et al., 2009)

4. Design Specification

In this study, the trigintaduonions algebra is a derived variation of the NTRU and implemented with python. In this section, the proposed new variation of the NTRU algorithm includes the key generation, encryption and decryption. For this key generation ;
W made use of 3 positive integers which are the N, p, q and the subsets of the L_f, L_g, L_m, L_h .

Key Generation :

For create the private and public keys for this algorithm an trigintaduonions sub algebra is generated off the the trigintaduonions algebra to generate the F And G

$F = x_0e_0 + x_1e_1 + x_2e_2 + x_3e_3 + \ldots\ldots + x_{30}e_{30} + x_{31}e_{31} \in 31$

$G = x_0e_0 + x_1e_1 + x_2e_2 + x_3e_3 + \ldots. + x_{28}e_{28} + x_{29}e_{29} + x_{30}e_{30} + x_{31}e_{31} \in 31$

This algebra must be invertible over the polynomial ring of R $= \dfrac{Z[x]}{x^N - 1}$

After this sub algebra are created it then creates the public and private keys ;

$F_q^{-1} = InverseFmodq$

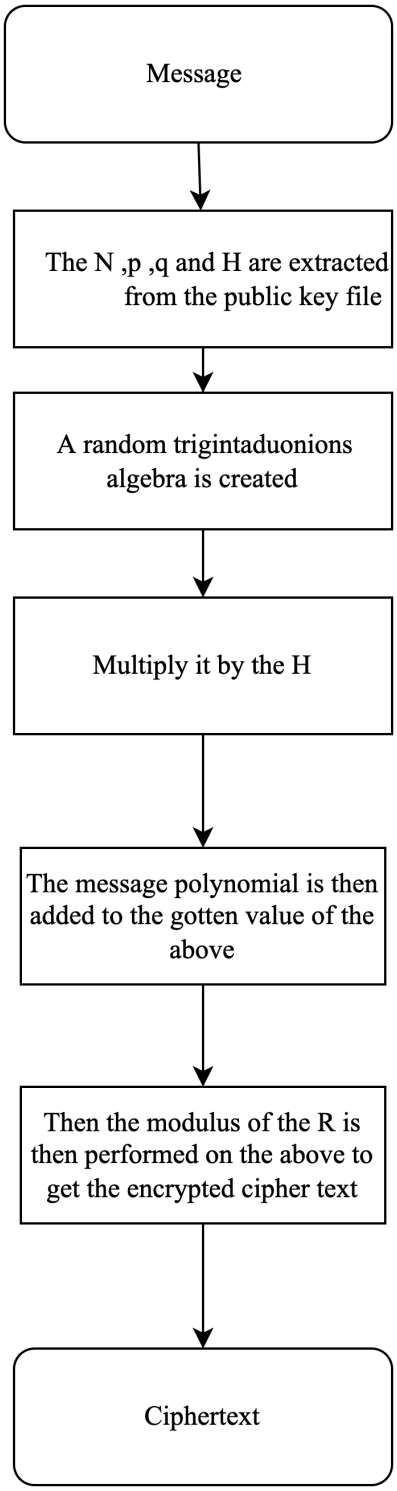
$F_p^{-1} = InverseFmodp$

The public key H = $F_q^{-1} \times p \times G$

The F and F_p^{-1} are the private keys that are kept and compressed into a file that would be used for the decryption phase of the algorithm while the N, p, q and H are compressed to be used as the public key file for the encryption of the message.

Encryption

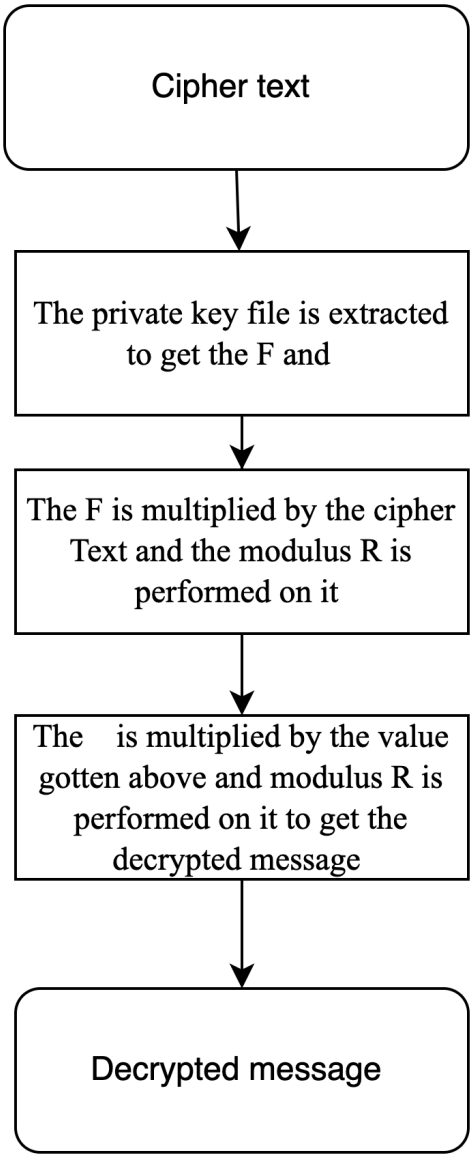
The given input includes the message and public key (.npz file) .



It could be mathematically written as $E = (H \times r + M) mod R$
Where r is the random algebra generated in step 1

Decryption

The Given input includes the encrypted message and the private key file (.npz)



It could be mathematically written as $M = (F_p^{-1} \times (F \times E) mod R) mod R$

5. Implementation

This section discusses the tools and languages that were used for the implementation of the new proposed encryption algorithm.

5.1 Python

Python is a multi-paradigm programming language with a high level of abstraction and dynamic typing.

5.2 Libraries

The standard library, which comes with Python, is a big set of pre-built and portable functionality. From text pattern matching to network scripting, this tool kit enables a wide range of application-level programming activities.

Sympy

SymPy is a symbolic mathematics Python package. Its goal is to develop into a full-featured computer algebra system (CAS) while keeping the code as basic as possible to make it intelligible and extendable.

Numpy

Python's NumPy package is the foundation for scientific computing. It includes a high-performance multidimensional array object as well as utilities for manipulating them.

Math

These routines do not handle complex numbers; instead, use the cmath module's functions with the same name. Because most users do not wish to study as much mathematics as is necessary to comprehend complex numbers, a difference is established between functions that support complex numbers and those that do not.

Counter

For counting hashable things, a Counter is a dict subclass. It's a collection in which the elements' counts are kept as dictionary keys and the elements' keys are stored as dictionary values.

5.3 Command shell

A shell is a user interface that allows the user to interact with operating system services. Shell takes human-readable commands from the user and translates them into kernel-friendly commands. It's a command language interpreter that can read instructions from keyboards or files and execute them. When a user signs in or launches a terminal, the shell is launched.

5.4 Sublime Text

Sublime Text is a source code editor that is available for purchase. It comes with built-in support for a variety of programming and markup languages. Plugins, which are often community-built and maintained under free-software licenses, allow users to extend the functionality of the system.

6. Evaluation

For this section, a comprehensive analysis of the results and main findings of these studies is presented. The effectiveness of this proposed algorithm when compared to other lattice-based algorithms like the NTRU and the STRU, is also highlighted here.

These algorithm effectiveness was evaluated on 2 key metrics:
1. Security complex analysis
2. Encryption Time

6.1. Security Complex Analysis

This metric of evaluation would be comparing two types of cryptography attacks against this proposed new algorithm and evaluating the chances of them executing. These two types of attack include the "meet the middle attack" and the "brute force attack"

6.1.1 Brute Force Attack

To use the brute force attack on the QSTRU, the complexity would need to be determined to easily compute the private key. For the brute force attack to be successful, many combinations of values would have to be tried till F*H (G*H^{-1}) produces a trigintaduonions. The total possible combination for this attack would have to run on various try values of F, and have to search in the following space given below ;

$$\left|L_f\right| = \binom{N}{d_f+1}^{32}\binom{N-d_f-1}{d_f}^{32} = \left(\frac{N!}{(d_f+1)!d_f!(N-2d_f-1)}\right)^{32}$$

Where L_f is the possible f polynomials

6.1.2 Meet in the middle Attack

Since the encryption message is E = (H * r + M) modR where r is the trigintaduonions generated. A meet in the middle attack can be used against the generated trigintaduonions, this is the same attack that could be used against the private key f.

For this new proposed algorithm the search space for the key message complexity for this attack would be $\sqrt{L_g} = \frac{N!^8}{d_g!^{32}(N-2d_g)!^8}$ while that opposed for the standard NTRU Would be $\sqrt{L_g} = \frac{N!}{d_g!(N-2d_g)!}$ (Hoffstein et al., 1998)

Where L_g is the one of the subset and the g polynomial for encryption

6.2. Encryption Time

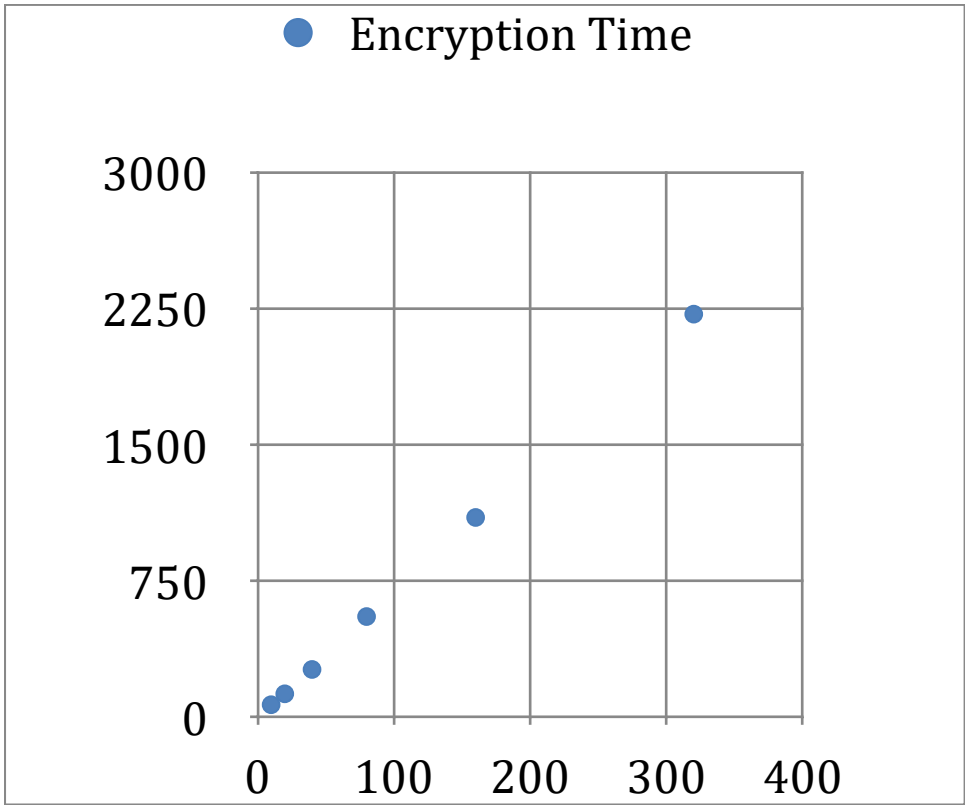
Encryption time is the metric that would be used to also measure and compare this proposed algorithm to the NTRU. I.e how much time it takes for the key and parameters to be generated and how long the encryption time takes for different file sizes.

Using public key values of N=587 p =3 and q = 256

File Size (KB)	Cipher Text size(KB)	Key Generation Time(s)	Encryption Time(s)
10	84	4.76	73.17
20	165	4.76	133.44
40	319	4.76	267.84
80	646	4.76	558.33
160	1324	4.76	1103.12
320	2648	4.76	2221.56

Table 2 : Encryption Time table

6.3. Discussion



Log plot of the QSTRU encryption time

From evaluating the proposed algorithm, it can be seen that possible space required for breaking into this algorithm is harder than compared to the NTRU, making it more secure. While doing the encryption of the data, using N as 587 the file size to encrypt and decrypt successfully using a classical computer as seen in the evaluation.

As opposed to the STRU algorithm with sedenion algebra which has16 sub loops that determine its structure making it run slower than the original NTRU using the time

complexity in table 3 below , the trigintaduonions also includes the sedenion as an aspect of complexity in its structure, making it run slower than the STRU which makes this a slower version as compare on the STRU algorithm while providing a higher message space in complexity.
With the presence of quantum computers, we could hypothetically say this slow speed would be irrelevant in encrypting the data as it is more secure than both of the algorithms, making it come down to a choice of trade-off with computing power and security.

From the table below we could see the QSTRU is better than the NTRU and STRU in the area of space complexity but in terms of time complexity the QSTRU is slower than both the STRU and the NTRU.

So while comparing the STRU QSTRU snd the NTRU :

	Space message complexity	**Time complexity**
QSTRU	$\left\|L_f\right\| = \binom{N}{d_f+1}^{32}\binom{N-d_f-1}{d_f}^{32} = \left(\frac{N!}{(d_f+1)!d_f!(N-2d_f-1)}\right)^{32}$	4.76
NTRU	$\left\|L_f\right\| = \binom{N}{d_f+1}\binom{N-d_f-1}{d_f} = \left(\frac{N!}{(d_f+1)!d_f!(N-2d_f-1)}\right)$	4.32
STRU	$\left\|L_f\right\| = \binom{N}{d_f+1}^{16}\binom{N-d_f-1}{d_f}^{16} = \left(\frac{N!}{(d_f+1)!d_f!(N-2d_f-1)}\right)^{16}$	4.54

Table 3: comparison for algorithms

7. Conclusion and Future Work

In the process of answering the research question of " how would a higher degree of complex algebra used as a variation of the NTRU improve the crypto system", the objective of creating a variation of the NTRU with the trigintaduonions algebra was achieved. The evaluations showed that the trigintaduonions algebra would improve the security of the cryptosystem, but at a higher cost of time complexity to enable encryption. Although this would not be a immense limitation in the time of the presence of the quantum computers, the present state complex algebra can be said to be efficient enough. This shows that a higher algebra increases security at a very high scale, but would also require a high amount of computing power to be very efficient.

For future work on this topic, deeper research would have to be carried out to improve the efficiency of this algorithm by increasing the amount of data that could be successfully encrypted and decrypted with this variation while also maintaining the high security of this algorithm.

References

Abo-Alsood, H. H., & Yassein, H. R. (n.d.). Design of an Alternative NTRU Encryption with High Secure and Efficient. 9.

Abo-Alsood, H. H., & Yassein, H. R. (2021). QOTRU: A New Design of NTRU Public Key Encryption Via Qu-Octonion Subalgebra. *Journal of Physics: Conference Series*, *1999*(1), 012097. https://doi.org/10.1088/1742-6596/1999/1/012097

Ajtai, M., & Dwork, C. (1997). A public-key cryptosystem with worst-case/average-case equivalence. *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, 284–293. https://doi.org/10.1145/258533.258604

Al-Saidi, N., Sadiq, A., & Majeed, A. (2016). CQTRU: A Commutative Quaternions Rings Based Public key Cryptosystem. *Library Hi Tech*, *34*.

Al-Saidi, N., & Yassein, H. (2016). *HXDTRU Cryptosystem Based on Hexadecnion Algebra*.

Bitru: Binary version of the ntru public key cryptosystem via binary algebra. (n.d.). Retrieved 2 November 2021, from

Buchmann, J. A., Butin, D., Göpfert, F., & Petzoldt, A. (2016). Post-Quantum Cryptography: State of the Art. In P. Y. A. Ryan, D. Naccache, & J.-J. Quisquater (Eds.), *The New Codebreakers: Essays Dedicated to David Kahn on the Occasion of His 85th Birthday* (pp. 88–108). Springer. https://doi.org/10.1007/978-3-662-49301-4_6

Butin, D. (2017). Hash-Based Signatures: State of Play. *IEEE Security Privacy*, *15*(4), 37–43. https://doi.org/10.1109/MSP.2017.3151334

Cawagas, R., Carrascal, A., Bautista, L., Sta. Maria, J. P., Urrutia, J., & Nobles, B. (2009). *The Subalgebra Structure of the Cayley-Dickson Algebra of Dimension 32 (trigintaduonion)*.

Chen, L., Jordan, S., Liu, Y.-K., Moody, D., Peralta, R., Perlner, R., & Smith-Tone, D. (2016). *Report on Post-Quantum Cryptography* (NIST IR 8105; p. NIST IR 8105). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.IR.8105

Courtois, N. T., Finiasz, M., & Sendrier, N. (2001). How to Achieve a McEliece-Based Digital Signature Scheme. In C. Boyd (Ed.), *Advances in Cryptology—ASIACRYPT 2001* (Vol. 2248, pp. 157–174). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-45682-1_10

Ding, J., & Petzoldt, A. (2017). Current State of Multivariate Cryptography. *IEEE Security Privacy*, *15*(4), 28–36. https://doi.org/10.1109/MSP.2017.3151328

Easttom, C. (n.d.). *An Overview of Quantum Cryptography with Lattice Based Cryptography*. 4.

Goldreich, O., Goldwasser, S., & Halevi, S. (1997). Eliminating decryption errors in the Ajtai-Dwork Cryptosystem. In B. S. Kaliski (Ed.), *Advances in Cryptology—CRYPTO '97* (pp. 105–111). Springer. https://doi.org/10.1007/BFb0052230

Hoffstein, J., Pipher, J., & Silverman, J. H. (1998). NTRU: A ring-based public key cryptosystem. In J. P. Buhler (Ed.), *Algorithmic Number Theory* (Vol. 1423, pp. 267–288). Springer Berlin Heidelberg. https://doi.org/10.1007/BFb0054868

Horowitz, M., Grumbling, E., National Academies of Sciences, Engineering, and Medicine (U.S.), National Academies of Sciences, Engineering, and Medicine (U.S.), National Academies of Sciences, Engineering, and Medicine (U.S.), & National Academies of Sciences, Engineering, and Medicine (U.S.) (Eds.). (2019). *Quantum computing: Progress and prospects*. the National Academies Press.

McEliece, R. J. (1978). *A public-key cryptosystem based on algebraic number theory*. Technical report, Jet Propulsion Laboratory, 1978. DSN Progress Report 42-44.

Lattice-based public-key cryptography. (n.d.). Retrieved 20 October 2021, from https://pqcrypto.org/lattice.html

Liu, D., Li, N., Kim, J., & Nepal, S. (2017). *Compact-LWE: Enabling Practically Lightweight Public Key Encryption for Leveled IoT Device Authentication* (No. 685). http://eprint.iacr.org/2017/685

Mohsen, A. W., Bahaa-Eldin, A. M., & Sobh, M. A. (2017). Lattice-based cryptography. *2017 12th International Conference on Computer Engineering and Systems (ICCES)*, 462–467. https://doi.org/10.1109/ICCES.2017.8275352

Peikert, C. (n.d.). *A Decade of Lattice Cryptography*. 90.

Ralph, M. (1987). *Advances in cryptology. 1987: Santa Barbara, Aug. 16-20, 1987*. CRYPTO.

Regev, O. (2009). On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, *56*(6), 1–40. https://doi.org/10.1145/1568318.1568324

Rieffel, E., & Polak, W. (2000). An introduction to quantum computing for non-physicists. *ACM Computing Surveys*, *32*(3), 300–335. https://doi.org/10.1145/367701.367709

Sendrier, N. (2011). Code-Based Cryptography. In H. C. A. van Tilborg & S. Jajodia (Eds.), *Encyclopedia of Cryptography and Security* (pp. 215–216). Springer US. https://doi.org/10.1007/978-1-4419-5906-5_378

Shor, P. W. (1994). Algorithms for quantum computation: Discrete logarithms and factoring. *Proceedings 35th Annual Symposium on Foundations of Computer Science*, 124–134. https://doi.org/10.1109/SFCS.1994.365700

SymPy. (n.d.). Retrieved 3 December 2021, from https://www.sympy.org/en/index.html

Thakur, K., & Tripathi, B. P. (n.d.). *STRU: A Non Alternative and Multidimensional Public Key Cryptosystem*. 18.

Yassein, H., & Al-Saidi, N. (2018). *BCTRU: A New Secure NTRUCrypt Public Key System Based on a Newly Multidimensional Algebra*.

Yassein, H. R., Abidalzahra, A. A., & Al-Saidi, N. M. (2021). *A new design of NTRU encryption with high security and performance level*. 080005. https://doi.org/10.1063/5.0042312

Yassein, H. R., Al-Saidi, N. M. G., & Farhan, A. K. (2020). A new NTRU cryptosystem outperforms three highly secured NTRU-analog systems through an innovational algebraic structure. *Journal of Discrete Mathematical Sciences and Cryptography*, 1–20. https://doi.org/10.1080/09720529.2020.1741218