IRTRU: An Algorithm for Post Quantum Cryptography

by

Kazi Md. Mahidul Islam
Exam Roll: Curzon Hall-249
Registration No: 2013-112-055
Session: 2013-14

Inzamamul Alam Munna
Exam Roll: Curzon Hall-271
Registration No: 2013-412-070
Session: 2013-14

A project submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY OF DHAKA

January 10, 2019

Declaration

We, hereby, declare that the work presented in this project is the outcome of the investigation performed by us under the supervision of Dr. Sarker Tanveer Ahmed Rumee,Assistant Professor, Department of Computer Science and Engineering, University of Dhaka. We also declare that no part of this project has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

.....

(Dr. Sarker Tanveer Ahmed Rumee)

Supervisor

Signature

.....

(Kazi Md. Mahidul Islam)

.....

(Inzamamul Alam Munna)

Abstract

We can tell that quantum computer become available in the near future. But there is no enough security system against the quantum computer although we already know that there exists an algorithm for breaking the classical approach by the quantum computer.That's why this is the high time to think over this issue. For this purpose we proposed a new algorithm in this project. There is some existing algorithm like NTRU for post quantum cryptosystem.but we want more secure because in NTRU there is some drawback. that's why our proposed algorithm:IRTRU is another cryptographic algorithm for post quantum cryptosystem. It's a public key cryptosystem. The way of encryption system is more complex than the existing ones. Its security is based on the presumed difculty of a lattice problem, namely, the shortest vector problem, a integrating function and also a rational polynomial function with some arithmetic calculation over polynomial and prime number. This Project report describes the *IRTRU* crypto-system and it's crypto-analysis.

Acknowledgements

All praise is to the Almighty, who is the most gracious and most merciful. There is no power and no strength except with Him.

Our deep gratitude goes to our thesis supervisor, Dr. Sarker Tanveer Ahmed Rumee, Assistant Professor, Department of Computer Science and Engineering, University of Dhaka, for his proper guidance in our research field. He has shared his expert knowledge gathered from working in the field over an extensive period, and has been an integral support in our thesis work by constantly keeping updates and urging us to something significant.

We want to thank our families and friends for their unwavering love and support. The opportunities that our parents have made possible for us determines the personalities we have built and the work that we produce today.

Lastly, we want to thank the Department of Computer Science and Engineering, University of Dhaka, its faculty, staff, and all other individuals related to the department. The department has facilitated us throughout our undergraduate program and subsequent thesis, and has also formed the base for our future endeavours.

Kazi Md. Mahidul Islam
Inzamamul Alam Munna
January, 2019

Contents

List of Algorithms

List of Tables

Chapter 1

Csystem(Cryptosystem) is the heart of the cyber security. If we want to protect our data from the attacker in the time of data passing through network we need to some algorithm and process to protect it otherwise some attacker will attack and one of them will successfully find out the data which will be very important data. And no one wants to discover his such important data to all.That's why we need Csystem to protect our data information.

1.1 Motivation

The all existing Csystem are valid only in classical computer.Assume when the Qcomputer are available in the world then what are the security become? Because all the PuKC(Public key Cryptosystem) which are valid in classical PC already proved by Shor that they are breakable . So what does about security? So information become insecure. That's why we must develop PQC(Post Quantum Cryptography) to secure our information in near future when the Qcomputer accessible. We already know that there is a Csystem known as NTRU, a new PuKC. NTRU is a Csystem for PQC which is already resolve this problem. the motivation of the proposed algorithm is to increase efficiency and to make data information more secure in the time of passing them by network.

1.1.1 Motivating Example

PuKC(RSA):

Factoring large integers is computationally intractable. This assumption is the base of *RSA*[1]. This is a valid assuption but only in classical computer.But not in Qcomputer.

Example:

1.1.1.1 A Con-create Example of Classical System(RSA):

Let two prime is 7 and 19. by calculating n and ϕ we get the value 119 and 108 respectively.Then choosing value e and k such that $ke = 1 mod\, \phi$, and we take e = 5 and k = 65.So according to RSA method the public key become (5,133) and Secret key become (7,19,65)

Encryption : Let The message is $M = 6$ Then we need to calculate the value of $C = M^e \,(mod\, n)$. So we find out the value of C, and it's become 62.

So The Message is encrypted here.

Decryption : To decrypt the message we need to calculate the value $C^k \,(mod\, n)$ which gives us original M.By calculating $62^{65} \,(mod\, 133)$. we findout the value of M which is 6.

So The Message is decrypted

Here is the counter example to break the classical algorithm in polynomial time.This algorithm is given by Shor[1].

1.1.1.2 Counter example to break Classical System(RSA):

Let Pick a number N then factorize it with two prime number by using Shor algorithm)

Pick a number y where y=5 and $x < N$ Then again pick a number $g = 18$ using QFT number where $F(x + g) = F(x)$ Now calculate GCD of $5^9 + 1$ and 133 and another GCD of $5^9 - 1$ and 133, After calculation we get the two GCD value is 7

and 19 which is our RSA's prime numbers. So the system is HACKED.

That's why need some better way to Csystem.We try to find out a new algorithm for the post quantum Csystem. Our Proposed algorithm is IRTRU which is a modifying version of NTRU.In our proposed algorithm we try to make Eprocess(Encryption process) more complex by choosing two random polynomial . Taking one polynomial and integrate it I times and taking another polynomial and find out two inverse polynomial function with modulo a and b which are two relatively prime number.This integrating process is our newly added functionality.And by this functionality the process become more complex and the attacker will face this complexity when he try to decrypt or find out something.

1.2 Organization of the Report

The project report is organized as follows:

Chapter 2 talks about the preliminary concepts that are relevant to the topic and discusses some of the state-of-the-art works that directly influence this study. In **Chapter 3**, the proposed algorithm is developed and an example is worked out. Performance evolution and Comparison of results of the proposed algorithm with existing algorithm is given in **Chapter 4**. In **Chapter 5**, the summary is provided as conclusion, and directions for future work is discussed. amsmath,amsfonts,amssymb,amsthm mathtools commath [sc,osf]mathpazo

Chapter 2

Background Study and Related Work

There is Two type of Csystem, asymmetric and symmetric. PuKC are known as asymmetric and PvKC are known as symmetric. Symmetric system are already breakable by the Qcomputer.So When Qcomputer are available we need some PQC that are protected when Qcomputer user attack.So we need some asymmetric Csystem.That why we need to know the existing Csystem.

2.1 Preliminary Concepts

Let us elaborate our discussion to better understand the concepts that lie at the heart of PQC.

2.1.1 Information Theory

2.1.1.1 Confusion

Confusion refers to creating the relationship between the ciphertext and the symmetric key as complex and involved as possible.

2.1.1.2 Diffusion

Diffusion refers to dissipating the statistical structure of plaintext over the bulk of ciphertext.

2.1.1.3 Security of a Crypto-system

There is such a thing as a Csystem that achieves perfect secrecy a Csystem in which the ciphertext yields no possible information about the plaintext (except possibly its length).Shannon theorized that it is only possible is the number of possible keys is at least as large as the number of possible messages. That means the key must be long as much as possible compare with messages.

A decent cryptographic calculation downplays this data. a decent cryptoanalyst abuses this data to decide the plaintext.

the entropy of a Csystem is a measure of the size of the key space,K, It is approximated by the base two logarithm of the number of keys:

$$H(K) = Log_2 K$$

2.1.2 Number Theory

It's a study of pure mathematics Csystem given principally to the examination.

2.1.2.1 Group

To construct cryptographic primitives group-based cryptography is a use of groups. A very general object is a group and most cryptographic schemes use groups in various way. The term group-based cryptography refers mostly to cryptographic protocols that use infinite nonabelian groups.

The integers mod n and the symmetries of a triangle or a rectangle are examples of groups. **A binary operation** or **law of composition** on a set G is a function G$\times G \to G$ that assigns to each pair (a,b) $\in G \times G$ a unique element $a \circ b$, or ab in G, called the composition of a and b. A group (G,$\circ$) is a set G

together with a law of composition (a,b)→ a∘b that satisfies the following axioms.

- The law of composition is associative. That is, (a∘b)∘c = a∘(b∘c) for a,b,c G.
- There exists an elemente G,called the identity element,such that for any element a ∈ G e∘a = a∘e = a.
- For each element a G, there exists an inverse element in G, denoted by a^{-1}, such that a∘a^{-1} = a^{-1} ∘a = e

A group G with the property that a ∘ b = b ∘ a for all a,b ∈ G is called abelian or commutative. Groups not satisfying this property are said to be nonabelian or noncommutative.

2.1.2.2 Ring

A nonempty set S become a ring by satisfying the following conditions:

- Maintain the associative law in addition and multiplication for all integers∈S.
- Maintain the distributive law in addition and multiplication for all integers ∈ S.
- Maintain the identity law in addition for all integers ∈ S.
- For every element a ∈ S, maintain the Inverse Property of Addition
- Maintain the commutative law in addition and multiplication for all integers ∈ S.

2.1.2.3 Field

Field is An integral domain in where we can find out a multiplicative inverse of each element.Addition, subtraction, multiplication and division all are possible in field.

2.1.2.4 Modular Arithmetic

Let n be a natural number. We say that two integers a and b are congruent modulo n if n divides $a - b$. We write this as

$$a \equiv b \mod n$$

.Here are the addition and multiplication tables of $\mathbb{Z}_5$. I have written the entries in the tables as a rather than $[a]_5$ to save clutter.

TABLE 2.1: Addition

0	1	2	3	4
1	2	3	4	0
2	3	4	0	1
3	4	0	1	2
4	0	1	2	3

TABLE 2.2: Multiplication

0	0	0	0	0
0	1	2	3	4
0	2	4	1	3
0	3	1	4	2
0	4	3	2	1

2.1.2.5 Extended Euclidean Algorithm(EEA)

EEA is the extended version of EA in computer programming and arithmetic . The formula is given below to follow the EEA and to find out the GCD of to number.

$$mX + nY = gcd(m, n).$$

This is a certifying algorithm,cause to satisfy this equation and divide the inputs in the same time gcd is the only one option.

Extended Euclidean algorithm also refers to a very similar algorithm for computing the polynomial greatest common divisor and the coefficients of Bzout's identity of two univariate polynomials.

Theorem:

For all positive integers m and n there exist integers X and Y such that

$$mX + nY = gcd(m, n).$$

The EEA[2] can be used to find X and Y.

In the time of computing Suppose that gcd(m,n) = 1. (That means they are relatively prime.) In this stage we can tell that m has a multiplicative inverse modulo n. That is, there exists an integer, which we call m^{-1}.

Here's the pseudo-code for the EEA:

$$
\begin{aligned}
&EEA(m,n)\{\\
&if(n == 0)\, return(m, 1, 0);\\
&(k1, X1, Y1) < -EEA(n, m\%n);\\
&k < -k1, X = X1;\\
&Y < -Y1 - (m\, div\, n) * Y1;\\
&return(k, X, Y);\\
&//returns(k, X, Y) such that k < -gcd(m, n)\\
&//div\, means\, integer\, division\, , k < -m * X + n * Y\\
&\}
\end{aligned}
$$

2.1.3 Cryptography Basic

Cryptography is the inquiry of sending and receiving system of data. Data authentication data affirmation data lifetime these all are related to cryptography system.Today's world cryptography is a most important topic in network security also in number theory and abstract algebra.

The message that an user send is known as **plaintext** .The encrypted message that an user get from the sender is known as **ciphertext**. The both formation are written with **alphabet**, comprising of **letters** or **characters**.The characters A, ..., Z and a, ..., z, in addition digits, accentuation stamps, and spaces all are use as alphabet or letters or as a character. A Csystem, or cipher, has two sections: **EProcess**, the system of changing data into ciphertext, and **Dprocess**, the system of getting the original data from ciphertext.

2.1.3.1 Private Key CryptoSystem(PvKC)

In **single** or **PvKC** we use the same key in both encrytion and Dprocess process. That's why the functionality of the EProcess and Dprocess are made in a way that no one can figure out the key. suppose f is a function and it will be encrypted and get $f^{-}1$ from that function. in the process that we get the inverse value is very secret to other and only it is known by the sender and receiver. That's why only receiver can decrypt it. And the PK is used to decrypt the data.

Example: The shift code used by Julius Caesar was the first and One of the most famous PvKC. The working process of this PvKC is first digitize the alphabet by letting $A = 00, B = 01, ...Z = 25$. The function of encoding system is given below,

$$G(X) = X + 4 \, mod \, 26;$$

that is, A $\rightarrow$ E,B $\rightarrow$ F,...,Z $\rightarrow$ D. The function of encoding system is become

$$G^{-1} = \text{X - } 4 \, mod \, 26 = p + 22 \, mod \, 26.$$

2.1.3.2 Public Key CryptoSystem(PuKC)

In the event that Csystem are utilized, any individual who realizes enough to encode a message will likewise realize enough to decode a captured message.W. Diffie and M. Hellman proposed PuKC in 1976, Which depends on the dissimilar key in two stage EProcess and Dprocess.That means sender use one key to EProcess and receiver use another key to Dprocess. This expels the prerequisite that the encoding key be kept mystery.Somebody who knows just the disordered mixture of key can't find out the the ordered mixture of key without restrictive calculation.

2.2 Related Works

There is a number of EProcess Dprocess like DES[4], AES[6], RSA[1] etc. to protect data from the attacker. Where DES and AES is Pvk block cipher EProcess Dprocess method and RSA is a Puk Eprocess Dprocess.

DES is a symmetric-key block cipher published by NIST.DES is a feistal cipher implementation is. 64-bits block size and 16 round feistal structure are used by DES.The effective key length is 56-bits although it's key length is taken 64-bits, since 8 of the 64 bits of the key are not used by the EProcess.

3DES are invented to increase security of the DES system.There are two Types of Triple DES known as 3-key Triple DES (3TDES[5]) and 2-key Triple DES (2TDES).

These days is AES is the more prevalent and widely used symmetric EProcess liable to be experienced .AES is 6 times faster than DES

Because of excess little size of key a swap is needed for DES With expanding processing power, it was viewed as defenseless against thorough key hunt assault. 3DES is introduced to prevent this disadvantage. The features of AES[6] are as follows

It provides full specification and design details and uses 128-bit data with 128 bits or 192 bits or 256 bits keys which is more stronger and faster than 3DES.

RSA (RivestShamirAdleman) is one of the first PuKC and is widely used for secure data transmission. In such a Csystem, the EProcess key is open and it is dissimilar from the Dprocess key which is kept secret. In RSA, this asymmetry is based on the factoring problem, the practical difficulty of the factorization of the product of two large prime numbers.

These are some Csystem and also there is existing algorithm to break them all. But for this purpose to break them we need to use Qcomputer as well as the knowledge of quantum cryptography. So when will the Qcomputers are available to most of the users, it is easy to them to break those method in polynomial time. So this is high time to discover new algorithm and also develop their security and efficiency against the Qcomputer which is known as post quantum cryptography.

2.2.1 PuKC RSA

RSA[1] is a popular PuKC and widely use.It is enough secure in classical computer and take less time to Eprocess and Dprocess. So If Qcomputer is not avilable RSA is one of the best choice to Eprocess and Dprocess.Here is the RSA algorithm.

Algorithm 1 RSA

Input: The Two prime number x,y

Output:

In Eprocess ouput = ciphertext C

In Dprocess output = Message M

Method:

Eprocess(x,y,M)

Dprocess(k,C)

Parameters: two large prime numbers.

Method:

1. Choose $n < -xy$, where x and y are distinct primes.
2. Calculate $\phi(n) < -(x-1)(y-1)$
3. Choose e & k,such as e with $gcd(e, \phi(n)) = 1$
4. Let one user is alif and another one is babu babu find out $k*e \equiv 1(\,mod\,\phi(n))$.
5. Babu makes (e,n) open and (x,y,k) private.
6. Alif convert message M as ciphertext $C < -M^e$ (mod n).
7. Babu again convert the ciphertext C by computing $M < -C^k\,(\,mod\,n)$.

2.2.2 Shor Algotrithm

Shor's algorithm, named after mathematician Peter Shor[3], is a quantum algorithmn (a algorithm that keeps running on a Qcomputer) for whole number factorization.

Shor algorithm is the counter example to break RSA algorithm in polynomial time but using Qcomputer. Without availability of Qcomputer there is no option to use Shor process. But when Qcomputer is available in the world then Shor algorithm is beneficial to the attacker to attack classical approach like RSA. Shor use a periodic function to implement his algorithm.Here is the procedure of Shor algorithm:

Algorithm 2 Shor's Algorithm

Input: Three numbers N and x and k where $k < N$

Output: Two prime number p,q

Method: Gcd($x^{\frac{r}{2}}$,N)

Subroutine: $F(t) = k^t \, mod \, N$

Parameters: random number k where $k < N$

Method:

1. At first take a random number k where $k < N$ and then find out the Gcd($x^{\frac{r}{2}}$,N).
2. If $Gcd(x^{\frac{r}{2}}, N) = 1$, then we can tell that this number is a nontrivial factor of N, so the process is done.
3. Not done yet? use the period-finding subroutine which is given below to find x, the period of the following function:
   $F(t) = k^t \, mod \, N$
4. This is the order x of a in the $(Zn)t$, is a very small positive integer value x which follows
   $F(t + x) = F(t)$
   $k^{t+x} \, mod \, N \in \, k^t \, modN.$
   (This is the quantum step of Shor)
5. If x is odd, go back to step 1.
   If $k^{x/2} \equiv -1 \, mod \, N$ go back to step 1
6. If $gcd(k^{x/2} + 1, N) and gcd(k^{x/2} - 1, N)$
   are both nontrivial factors of N. Then we are done with our solution p,q which our prime factors.

2.2.3 PuKC Elliptic Curve(ECC)

ECC[7] is an PuKC.Each client has two keys in ECC,one is Puk and another is Pvk. In EProcess Puk is utilized and for Dprocess Pvk is utilized . The PvK is

also used for signature generation.

2.2.3.1 Key Generation of ECC

To produce both PuKs and PvKs a algorithm is utilized and it is the most vital advance in which.With the assistance of recipient's Puk sender create ciphertext and beneficiary create the original data using its Pvk.

- Stage 1.At first an arbitrary dx number is choosen by the encrypter between the range $[1, l-1]$.This is the Pvk of of the encrypter.
- Stage 2.By this Equation $Px = dx * F$, sender creates the general Puk.
- Stage 3. Correspondingly a Pvk dy is choosen by the decrypter and produces its Puk by this equation, $Py = dy * F$.
- Stage 4. $pvk = dx * Py$ is a security key is producing by the sender and additionally the security key $puk = dy * Px$ is created by the beneficiary .

2.2.3.2 EProcess System of ECC

Suppose someone called sender wants to sending data information to a specific person called receiver. He can use the ECC Eprocess. Thta's why he should follow this step or algorithm:

- Step 1.On the elliptic curve let DI has any point MM
- Step 2.A random number kK is selected by the sender from the range $[1, l-1]$
- Step 3.By the pair of $points(P1, P2)$ the cipher texts will be generated where,
  $P1 = kK * F$
  $P2 = MM + (kK * F)$

2.2.3.3 Dprocess System of ECC

following steps are performed to retrieve the ciphertext :

- Step 1.At first the product of $P1$ and its Pvk is computed by the receiver.
- Step 2.Then subtracts this product from the second point $P2$ $MM = P2 - (dy * P1)$ by the receiver

MM is the original data information that was sent by the sender.

2.2.4 PukC NTRU

in 1996 by Hoffstein, Pipher and Silverman the NTRU[8] Csystem is presented as a fast PukC. It can protect data information although the attacker attacks the ciphertext by using a Qcomputer and is categorized as a PQC. factoring large composite integers or computing discrete logarithms are the base of the security of RSA[1], ECC[7] and El Gamal.Someone name Shor[3] proved that in polynomial time, Qcomputer can factor integers and compute discrete logarithms. That's why all the previous Csystem like RSA, ECC and El are easily breakable by Shor procedure. So when the Qcomputer available in large scale, many efforts have been deployed to ensure the cryptographic protocols visibility in future. Hence, some PukCsystem have been developed that are believed to be resistant to quantum computing based attacks such as the NTRU Csystem. An interesting advantage of NTRU over traditional puKCsystem based on factoring or discrete logarithm is its potential resistance to Qcomputers. This makes it a promising alternative to the more established PukC. that's why one of the prominent PQC is considered as NTRU . The security of NTRU is related to a very hard problem. Which is known SVP used for lattice reduction. It is conjectured that to solve this problem there is no polynomial time algorithm. .

2.2.4.1 Description OF NTRU

The NTRU EProcess scheme

There is three integer parameters (N, a, b) in NTRU.The arithmetic of NTRU depends on this N,a,b . Let the ring of integers is denoted by $\mathbb{Z}b = \mathbb{Z}/b\mathbb{Z}$ modulo q.in the ring of truncated polynomial, the all operation of NTRU are held. $a = \mathbb{Z}b[T]/(T^N - 1)$

Let in this ring, f is a polynomial defined by its coefficients in the base $1, T, T^2, ..., T^{N-1}$ as

$$f = (f_0, f_1, ..., f_{N-1} = f_0 + f_1 T + ... + f_{N-1} T^{N-1})$$

The addition of two polynomials are defined as followed,

$$f + r = (f_0 + r_0, ..., f_{N-1} + r_{N-1}$$

The multiplication of two polynomial, noted " * " is given below,

$$f * r = m = (m_0, m_1, ..., m_{N-1}) with\, m_k = \sum_{i+j*k\,(mod\,N)} f_i r_j$$

The Euclidean norm or the length of a polynomial $r = (r_0, r_1, ..., r_{N-1})$ is defined as

$$\|r\| = \sqrt{\sum_{i=0}^{N-1} r_i^2}$$

Different descriptions of **NTRUE**ncrypt, and different proposed parameter sets, have been in circulation since 1996.In NTRU here setup six public integers $N, a, b, d_{ff}, d_{gg}, d_{rr}$ and four public spaces $L_{ff}, L_{gg}, L_{mm}, L_{rr}$.

- N is a prime. To prevent lattice attacks it will be chosen as sufficiently large .
- a , b are relatively prime numbers.
- a is much smaller than b.

- $L_{ff} = S(d_{ff})$ is a set of small polynomials. The PvK are selected from this set.
- $L_{gg} = S(d_{gg})$ is a another set of small polynomials and similar to the previous one.The other PvK are selected.
- $L_{mm} = \mathbb{Z}_a[T]/(T^N - 1)$ is the plaintext space. It is a set of polynomials $m \in \mathbb{Z}_a[T]/(T^N - 1)$ that represent encryptable messages.
- $L_{rr} = S(d_{rr})$ is a set of polynomials from which the blinding value used during EProcess is selected.

The key generation, EProcess and Dprocess primitives are as follows:

2.2.4.2 Key generation of NTRU

- First ff is Randomly chosen polynomial. here $ff \in Lff$ such that ff is invertible in a modulo a and modulo b.
- Compute $ff_a \equiv ff^{-1} \,(mod\, a)$ and $ff_b \equiv ff^{-1} \,(mod\, b)$.
- $gg \in L_{gg}$ is another randomly chosen polynomial.
- Compute $w \equiv gg * ff_b \,(mod\, b)$.
- Publish the Puk (N, w) and the set of parameters $a, b, L_{ff}, L_{gg}, L_{rr}$ and L_{mm}.
- Keep the Pvk (ff, ff_b).

2.2.4.3 EProcess of NTRU

- $mm \in L_{mm}$] is a polynomial which is represent the data information.
- $rr \in L_{rr}$ is a Randomly chosen a polynomial.
- produced mm with the Puk (N, w) using the rule $e \equiv a{*}rr{*}w{+}mm \,(mod\, b)$.

2.2.4.4 Dprocess of NTRU

- Here s ≡ ff * e (mod b) which computes by the receiver.
- this s is transform into a polynomial with a centering procedure in a range or interval:
  $\big[-\frac{b}{2}, \frac{b}{2}\big[$.
- Then calculate $mm \equiv ff_b * s \,(mod\, a)$.

The Dprocess process is correct if the polynomial $a * rr * gg + ff * mm \,(mod\, b)$ is truly equal to $a * rr * gg + ff * mm \in \mathbb{Z}[T]/(T^N - 1)$, that is without using modulo b. We have

$$\begin{aligned} s &= ff * e(mod\, b) \\ &= ff * (a * rr * w + mm)(mod\, b) \\ &= ff * rr * (a * gg * ff_b) + ff * mm(mod\, b) \\ &= a * rr * gg * ff * ff_b + ff * mm(mod\, b) \\ &= a * rr * gg + ff * mm(mod\, b). \end{aligned}$$

Hence, if $s = a * rr * gg + ff * mm \in \mathbb{Z}[T]/(T^N - 1)$, then $s * ff_a \equiv (a * rr * gg + ff * mm) * ff_a * mm \,(mod\, a)$

The Dprocess never fails, If the parameters are chosen correctly. A sufficient condition for this is to choose a much smaller than b.

2.2.4.5 An example of NTRU EProcess

To illustrate the NTRU[8] scheme, consider the following parameters

$$N = 11.$$
$$a = 3.$$
$$b = 61.$$
$$ff = -T^{10} - T^8 - T^6 + T^4 + T^2 + T + 1.$$
$$gg = -T^9 - T^8 - T^6 + T^4 + T^2 + 1.$$
$$mm = T^7 - T^4 + T^3 + T + 1.$$
$$rr = -T^9 + T^7 + T^4 - T^3 + 1.$$

Then, we get

$$ff_a = T^9 + T^7 + T^5 + 2T^4 + 2T^3 + 2T^2 + T.$$
$$ff_b = 45T^{10} + 49T^9 + 26T^8 + 40T^7 + 53T^6 + 47T^5 + 21T^4 + 24T^3 + 60T^2 + 32T + 31$$
$$w = 11T^{10} + 49T^9 + 25T^8 + 46T^7 + 28T^6 + 53T^5 + 31T^4 + 36T^3 + 32T^2 + 5T + 50.$$
$$e = 11T^{10} + 46T^9 + 52T^8 + 35T^7 + 30T^6 + 26T^5 + 35T^4 + 32T^3 + 18T^2 + 56T + 28.$$

Then, computing $s = ff * e \,(mod\, b)$ and centering modulo b, we get,

$$s = 58T^{10} + 60T^9 + 60T^8 + 4T^7 + 56T^5 + 6T^4 + 55T^2 + 3T + 6.$$
$$s = -3T^{10} - T^9 - T^8 + 4T^7 - 5T^5 + 6T^4 - 6T^2 + 3T + 6.$$

Finally, computing $fp * a \,(mod\, p)$ and centering modulo p, we get,

$$ff_a * s = T^7 + 2T^4 + T^3 + T + 1 \,(mod\, a)$$
$$ff_a * s = T^7 - T^4 + T^3 + T + 1$$

and we get mm which is our original data information.

2.3 Summary

NTRU is better algorithm than other Csystem like RSA[1] ECC[7] in PQC. As a result,From NTRU[8] , we are introduced with some algorithm and mathematical background such as lattices, SVP[9] and CVP[10], LLL-Algorithm[12], GVH and CPR[11]. Also it described the NTRU Csystem parameters, Pvk, Puk, EProcess and Dprocess. We also introduced with DES[4], 3DES[5], AES[6], RSA some other Csystem.

Chapter 3

The Proposed Approach

In the previous chapter we discussed about some existing CS. Also we come to know some mathematical proof and algorithm . This study is very essential important to implement our proposed system that is IRTRU a new algorithm over NTRU[8] which is modifying version of NTRU.In this chapter we will discuss about IRTRU CS. How IRTRU work, the system of key generation also the Eprocess and Dprocess.

3.1 Mathematical Background

The security of the IRTRU CS is identified with the difculty of finding a short vector of a given Lattice and rational function with integrate a function for increasing it's power. Henceforth, in this part we will survey lattices and their properties. What's more, the LLL-algorithm[12], GVH(GVH) and the CPR(CPR) are additionally audited.

3.1.1 Lattices

In this section we review the concepts from the theory of lattices that are needed to describe the IRTRU CS.

Clarity2.1. Let $d_1, d_2, d_3, ..., d_k$ be a lot of straight autonomous vectors in $\mathbb{ZZ}^N$. The lattice $\mathbb{LL}$ generated by $d_1, d_2, d_3, ..., d_k$ is the arrangement of direct mixes $d_1, d_2, d_3, ..., d_k$ with coefcients in $\mathbb{ZZ}$.

$$\mathbb{LL} = \left\{c_1d_1 + c_2d_2 + c_3d_3 + \ + c_kd_k | c_1, c_2, ..., c_k \in \mathbb{ZZ}\right\}$$

The integers R and D are disparately called the Rank and Dimension of $\mathbb{L}$. Lattice $\mathbb{LL}$ is called full rank when R = D and we assume full rank from here on. We arrange the vectors $d_1, d_2, d_3, ..., d_k$ as rows of a matrix B, which is then called or said the generator matrix of the lattice. The same lattice can have different basis. Assume $d_1, d_2, d_3, ..., d_k$ is a basis for a lattice $\mathbb{LL}$ and another collection of vectors is $p_1, p_2, p_3, ..., p_k \in \mathbb{L}$. Then we can write the p_j as a linear combination of $d_1, d_2, ..., d_k$ as follows:

$$p_1 = c_{11}d_1 + c_{12}d_2 + \ + c_{1k}d_k,$$

$$p2 = c_{21}d_1 + c_{21}d_2 + \ + c_{2k}d_k,$$

$$...$$

$$p_k = c_{k1}d_1 + c_{k2}d_2 + \ + c_{kk}d_k,$$

Note that all of the c_{ij} coefcients are integers we can discuss about fractional coefficient in future work.

3.1.2 The Shortest Vector (SVP) and Closet Vector Problem (CVP)

In this section we dene and discuss two fundamental computational problems related to lattices and this are the base of our theory about our proposed algorithm.

Clarity2.3. (SVP[8]): Assume a lattice $\mathbb{LL}$:,Find a shortest nonzero vector in it, i.e, nd a nonzero vector $d \in \mathbb{LL}$ such that the Euclidean norm $||d||$ is the minimum compared to others.

Clarity2.4. (CVP[9]): Assume a lattice and a vector $V \in \mathbb{RR}^n$ that is not in LL,Find a vector $d \in \mathbb{LL}$ that is closest to V, i.e, a vector $d \in \mathbb{LL}$ that minimizes the Euclidean norm $||V - d||$. Euclidean norm is the only distance between two vectors or two points.

Note that there might be more than one nearest vector in a cross section and likewise more than one most limited vector in a grid. Accordingly, we search for "a" nearest vector, not "the" nearest vector. For instance, in $\mathbb{ZZ}^2$, vectors (0,±1) and (±1,0) are the briefest vectors. Both the SVP and CVP[10] are difcult problems. Additionally, CVP is NP-hard and SVP[9] is NP-hard under a specific randomized decrease speculation. The variation of the shortest vector problem and the closest vector problem are the shortest basis issue, estimated shortest vector problem and approximate closest vector problem.

Clarity2.5.Shortest basis Problem SBP: Find a basis $d_1, d_2, d_3, ..., d_k$ for a lattice that is shortest. For example, we require that greatest $||d_i||$ or the $\sum_{j=1}^{N} ||d_i||^2$ should be minimized.

Clarity2.6. Approximate Shortest Vector Problem(apprxSVP): Let $\alpha(k)$ be a function of K. Given a lattice $\mathbb{LL}$ of dimension D, Find a nonzero vector that is less than or equal to $\alpha(k)$ multiplies by a shortest vector, that means if $d_{shortest}$ is a shortest vector in $\mathbb{LL}$, Find a nonzero vector $d \in \mathbb{LL}$ such that

$$||d|| \leq \alpha(k)||d_{shortest}||.$$

Note that individual choice of function $\alpha(k)$ give us a different apprxSVP. It is obvious that apprxSVP is SVP[9] if $\alpha(k) = 1$.

For any ordered lattice basis $BB = \{d_1, d_2, d_3, ..., d_N\} \in \mathbb{RR}^N$ the set of corresponding Gram-Schmidt orthogonal vectors $BB = d_1^*, d_2^*, d_3^*, ..., d_N^* \subseteq \mathbb{RR}^N$ is state by

$$d_i^* := d_i - \sum_{j=1}^{j-1} p_{ij} d_j^* \, for \, 1 \leq j \leq N$$

where $p_{ij} := \langle d_i, d_j^* \rangle / \langle d_j^*, d_j^* \rangle \, for \, 1 \leq i \leq N$. If we want to solve the SVP from a given lattice and basis BB, we are drawn to use the Gram-Schmidt orthogonalization process above to find an orthogonal basis $BB^{'}$. Given an orthogonal basis, then the shortest non-zero vector is a basis vector.

epistle that in the orthogonalization procedure, we multiply the basis BB by a matrix that is invertible but not all entries are integers. (But in case of our algorithm,we must work with integer) Thus, the lattice $LL^{'}$ generated by$BB^{'}$ is not the same as the original given lattice generated by BB.

Comment: Not every lattice has an orthogonal basis; for consequence, we have to nd a intellectual method that works for all of these lattices. In 1982, A. Lenstra, H. Lenstra and Lovasz in created the LLL-Algorithm for this purpose. In order to apply the LLL-Algorithm, we need to know the LLL-reduced and LLL-reduced basis vectors. The following denition and propositions are in of $A.Lenstra, H.LenstraandLovasz.$

Clarity2.7. Let $LL(d_1, d_2..., d_N) \in \mathbb{RR}^N$ then the basis vectors are $d_1, d_2, ..., d_N$ called LLL-reduced with $\frac{1}{4} < \Delta < 1$, if $|p_{ij}| \leq \frac{1}{2}$ where $1 \leq j \leq N$,

$$||d_{i-1}^*||_2^2 \leq ||d_i^* + p_{ii-1}d_{i-1}^*||_2^2 \, where \, 1 < i \leq N$$

3.1.3 Gauss Volume-Heuristic GVH

In this section, we will discuss the GVH for our security purpose which will discuss later in **chapter 4.**

Given a lattice LL and $SS \subseteq span(LL)$ a measurable set, the expected value is $E[(SS \cap LL)] = \frac{vol(SS)}{detLL}$

there exists some principal locale. Consequently, the normal number of grid focuses in SS is equivalent to the volume of the central area $vol(LL)$ divide k the volume of SS.

Theorem2.1. Let $\mathbb{BB}_R$ be a ball with radius R and center 0 in $\mathbb{R}^N$. Then the volume of $\mathbb{BB}_R$ is

$$vol(\mathbb{BB}_R) = \frac{N/2R^N}{\alpha(1+\frac{N}{2})}$$

where α denoted the volume of the ball $\mathbb{BB}_R$ in $\mathbb{R}^N$ is approximation given by

$$vol(\mathbb{BB}_R)^{\frac{1}{N}} \approx \sqrt{\frac{2\pi e}{N}}R$$

clarity2.8. Let LL be a lattice in$\mathbb{RR}^N$. Then the Gaussian expected shortest length is

$$ss \approx \sqrt{\frac{N}{2\pi e}}(detLL)^{\frac{1}{N}}$$

This Gaussian heuristic states that a shortest nonzero vector in a arbitrary lattice will fulfill

$$||d_{shortest}|| \approx ss$$

3.1.4 Convolution Polynomial Ring CPR

The CPR is a unique ring that is utilized by the IRTRU Csystem. Thus, we will survey them now

clarity2.9. For a positive integer n,the CPR is the quotient ring

$$RR = \frac{Z[t]}{(t^n-1)}$$

Similarly, the CPR (modulo b) is also the quotient ring

$$RR_q = \frac{(Z/_bZ)[t]}{(t^n-1)}$$

Every element of $R\,or\,R_q$ has a novel portrayal of the frame $d = d_0 + d_1t + ... + d_{n1}t^{n-1}$ with the coefficients in $Z\,or\,Z/_bZ$ respectively. Since in the ring RR we know that $t^n \equiv 1$, the multiplication of two elements $d, s \in RR$ is given by $w = d.s(mod\,t^n-1)$. Note that it is much less demanding to figure in the ring $RR\,and\,RR_b$at that point it is in the general polynomial remainder rings on the grounds that the polynomial has a straightforward shape t^n-1 has a straightforward form. This clarification bodes well in light of the fact that at whatever point we mod out by t^n-1, we simply require t^n to equal 1. A coefcient w_k of the product is calculated as

$$w_k = \sum_{i+j=k \, mod \, n} d_i * s_i$$

$$= \sum_{i=0}^{k} d_i.s_{k-1} + \sum_{i=k+1}^{n-1} d_i.s_{n+k-1}$$

Note that a vector $d = (d_0, d_1, ..., d_{n-1})$ represents the polynomial $d = d_0 + d_1 t ... + d_{n-1} t^{n-1} in \, RR$

For example:

$$(t^5 + 2t^2 + 1) * (3t^4 + t^2) = 3t^9 + t^7 + 6t^6 + 5t^4 + t^2$$

$$= 3t^2 + 1 + 6t^6 + 5t^4 + t^2$$

$$= 6t^6 + 5t^4 + 4t^2 + 1$$

in the polynomial ring $ZZ[t]/(t^7 - 1)$.

3.2 The Proposed Algorithm:IRTRU

In this section, Our Csystem:*IRTRU* is based on polynomial function in a polynomial ring. We try to reduce the polynomial in a lattice and adding integration to the polynomial to reduce the attack on a polynomial time.

3.2.1 Parameters of:*IRTRU*

The Basic Parameters of our Csystem is I,N,l,u,M,ϕ,f,g,P.

the details of our parameters:

I Number of integration,preferred to Prime.

N Degree of the polynomial used in the polynomial rings,It also preferred to prime. It will improve security of our Csystem.

l & u are the relatively prime

where $u > l \,\&N > u \,\& \, N > I > u$

P is the Public key

f & g are the two randomly chosen function which has inverse.

ϕ is also a random polynomial which is discarded after use.

3.2.2 Private key of our Csystem

To create the private key bob choosing two random polynomial f g . where f & g in invertible module of $l \,\& \, u$ .which is $f_l \,\& f_u$.
f_u is reduce the co-efficient invertible of l in $[X^N - 1]$
f_l is reduce the co-efficient invertible of u in $[X^N - 1]$.

g_u is reduce the co-efficient invertible of l in $[X^N - 1]$
g_l is reduce the co-efficient invertible of u in $[X^N - 1]$.
The equation of f_u & f_l & g_u & g_l is :

$$f_l * f \equiv 1 \, (mod \, l)$$

$$f_u * f \equiv 1 \, (mod \, u)$$

$$g_l * g \equiv 1 \, (mod \, l)$$

$$g_u * g \equiv 1 \, (mod \, u)$$

the private parameters set are $(f, f_l, f_u, g, g_l, g_u)$

3.2.3 Public key of our Csystem

Bob Choose the public key for this.At first Bob choose g and computes g_l and integrate this g_l polynomial with the number I the integrating method for this is :

$$\Gamma \longrightarrow [{}_I\!\int \ \ldots \ {}_1\!\int \ g_l \, dx] \, mod \, u$$

the Bob computes the public key by a rational function which.

$$P \equiv \ l * f_l * \tfrac{\Gamma}{f_u} (mod \, u)$$

The public parameter set are (I, N, l, u, P)

3.2.4 Encryption of our Csystem

Alice Choose a message M which co-efficient always lies between $-\frac{1}{2}(p-1) \, and \, \frac{1}{2}(p-1)$. Now Alice choose a random polynomial ϕ, and Encrypt his message. By this formula.

$$\begin{aligned} E &\equiv \ \phi * P \ + M \, (mod \, u) \\ E &\equiv \ \phi * l * f_l * \ \frac{\Gamma}{f_u} + M \, (mod \, u) \\ E * f_u &\equiv \ \phi * l * f_l * \Gamma \ + \ M * f_u \, (mod \, u) \end{aligned} \tag{3.1}$$

3.2.5 Decryption of our Csystem

Suppose that Bob has received the ciphertext from Alice and wants to decrypt it using his private key f. To do this efficiently, Bob should have precomputed the polynomial D . In order to decrypt E,

$$D \equiv \ E * f_u \, (mod \, u)$$

3.2.6 Decryption Process Validity check

During the decryption process, we use the public key P and the private key pair(f, f_u, f_l). The encrypted message is $E \equiv \phi * P + M \,(mod\, u)$, and it is the sum of the original message M with the scaled version of the public key P and a rational Function $\frac{\Gamma}{f_u}$. Out of the parameter, Bob only has the value of public key P, which he has to use mathematical methods to recover the message.
Note that the coefcient of the polynomials ϕ, Γ, M and modulus l are very small compared to the coefcient of the modulus u. Now the encrypted message multiply the private key f_p to eliminate. Recall that all of the computation are performed in the CPR. If encrypted message is

$$\begin{aligned} E &\equiv \phi * P + M \,(mod\, u) \\ E &\equiv \phi * l * f_l * \frac{\Gamma}{f_u} + M \,(mod\, u) \\ E * f_u &\equiv \phi * l * f_l * \Gamma + M * f_u \,(mod\, u) \end{aligned} \tag{3.2}$$

multiply the encrypted message by the private key f_u

$$\begin{aligned} D &\equiv E * f_u \,(mod\, u) \\ D &\equiv \phi * l * f_l * \Gamma + M * f_u \,(mod\, u) \end{aligned} \tag{3.3}$$

Note that the modulus u is big than the private key f,f_u. Thus the value of D, containing polynomials and a scalar, is very small compared to the modulus q.
Consider this last polynomial. For appropriate parameter choices, we can ensure that (almost always) all of its coefficients lie between (-u/2,u/2], so that it doesn't change if its coefficients are reduced modulo u. This means that when Bob reduces the coefficients of $f_u * E \,modulo\, u$ into the interval from (-u/2,u/2], he recovers exactly the polynomial

$$\phi * l * f_l * \Gamma + M * f_u \text{ in } R^N$$

Reducing D modulo l then gives him the polynomial

$$M * f_u$$

and then multiplying by f. This produce.

$$M \text{ because } f_u * f \equiv 1$$

3.3 Summary

The proposed *IRTRU* algorithm provides the complete details of how Encryption and Decryption works in an efficient way for text. It will generate results faster than RSA and others key, and accurate results as well. It will also consume less memory as we are dealing with the modulus to reduce it's coefficient. This can also be applied in real life .

Chapter 4

Attacks and performance analysis of IRTRU

4.1 Introduction

In this chapter, we will introduce the *Another PvKC attack*. We also examine the *brute-force(all possible keys) attack* and an improved attack called *meet-in-the-middle attack*. Furthermore, we will analyze why we should not send a message multiple times with the same public key h in the *multiple message attack*. Finally, we will show the interesting attack on IRTRU called the *latticeBased attack*.

4.1.1 another private key attack

The alternate PvKC f^a can be used to decrypt the same message as f. Let f^a be any rotation of the private key f, then $f^a = x^j * f$ for $j \in \{1, 2, ..., N-2\}$.

Here,

$$
\begin{aligned}
f_u^a &= x_j * f_u. \\
f_l^a &= x_j * f_l. \\
where\, j &= 1, 2, 3, ..., N-2. \\
P &= l * f_l * \frac{\tau}{f_u} (mod\, u). \\
so,\ P * f_u (mod\, u) &= l * f_l * \tau. \\
then\, \tau &= \frac{P * fu}{l * fl} (mod\, u) \\
&= \frac{P * x_j * f_u}{l * x_j * f_l} (mod\, u) \\
&= \frac{P * f_u^a}{l * f_l^a} .(mod\, u) \\
so\ in\ decryption & \\
D &\equiv \phi * l * f_l * \tau + M * f_u\, (mod\, u). \\
&\equiv M * fu\, (mod\, u). \\
multiplybothsidebyx^j so,\ D * x_j &\equiv M * x_j * f_u\, (mod,\ u). \\
&\equiv M * f_u^a\, (mod\, u).
\end{aligned}
$$

So the attacker can decrypt M(message).

4.1.2 Brute force attack

An attacker can get the PvK(private Key) by trying all possible $f \in Ł_f$, and $g \in Ł_g$. Similarly an attacker can get the message by trying all possible $\phi \in L_\phi$.In general, this attack will cut the search time by square root. Thus, the security level is. here L_f,L_g is the set of all private keys. and L_ϕ is the set of blinding keys. L determine the security of keys and message

$$
\begin{aligned}
\sqrt{\#L_f} &= \sqrt{({}^nC_{d_f} * (n - d_f) C_{d_f})} \\
where\, d_f &= number\, of\, 1's\ in\, f(x).
\end{aligned}
\tag{4.1}
$$

$$\sqrt{\#L_\phi} = \sqrt{({}^nC_{d_\phi} * (n-d_\phi)C_{d_\phi})}$$
$$where\, d_\phi = number\, of\, 1's\, in\, \phi(x). \tag{4.2}$$

4.1.3 Meet in the middle attack

In NTRU ,This attack can likewise be utilized against this CS utilizing a similar argument on the degree of the rational coefficient of polynomials. This attacks needs a ton of capacity limit and cut the search time by the typical square root. Hence it means that the set of possible gg and rr has to contain at least 2^{160} elements in order to obtain a security of 2^{80}.

but in IRTRU , we claim that the meet in the middle attack will sufficiently complicated rather than NTRU. because we use three private keys in PuK creation process and the possibility of the middle attack is $\frac{1}{6}$. and more complicated to find one key with this. the obtaining security approximately of 2^{240} . In the future,we work with IRTRU meet in the middle attack with proper calculation.

4.1.4 multiple message attack with same public key

The fundamental reason for this attack is to demonstrate to us that it's anything but a smart thought to communicate something specific a few times with a similar PuK P . Alice communicates something specific M a few times to Bob,with a similar Puk, however she changes the arbitrary polynomial $\phi \in L_{phi}$ each time. In other words, we have n diverse irregular polynomial which we indicate by ϕ_j where $j = 1, 2, ..., n-1$ and afterward Alice ascertains the distinctive cipher text which are given by

$E_j \equiv P*\phi_j + M\,(mod\, u)$ for $j = 1, 2, ..., n-1$ and after that she sends them to Bob. Thus, Eve captures these diverse cipher text and gets all E_j where $j = 1, 2, ..., n-1$ and calculates

$$E_i - E_j * P^{-1}\,(mod\, u)\, where\, 1 \leq i < j \leq n-1$$

and gets

$$\phi_i - \phi_j \,(mod\, u)\, where\, 1 \leq\ i\ < j \leq\ n-1$$

Note that the coefficients of $\phi_i - \phi_j$ range from -2 to 2 this is must obvious, thus Eve recovers $\phi_i - \phi_j$ with max probability. Presently given us a chance to examine the conceivable outcomes for some specific coefcient of $\phi_i - \phi_j$, say the i^{th} coefcient. If we let

$\Theta = the\, i^{th}\, coefficient\, of\, \phi_i$,

$\eta = the\, i^{th}\, coefficient\, of\, \phi_j$,

$\Delta = the\, i^{th}\, coefficient\, of\, \phi_i - \phi_j$

The possibilities for $\Delta = \Theta - \eta$ are listed in the following table:

TABLE 4.1: $\phi_i - \phi_j$ *is the coefficient of polynomial*

$\Delta = \Theta - \eta$	-1	0	1
-1	0	-1	-2
0	1	0	-1
1	2	1	0

On the off chance that $\Delta = 2$ (respectively $\Delta = -2$), then Eve derives that $\eta = -1$ (respectively $\eta = 1$). Therefore, Eve will have the capacity to recuperate around $\frac{2}{9}$ of the coefficients of ϕ_i. Moreover, if $\Delta = 1$ (respectively $\Delta = 1$), Eve concludes that $\eta = 1, 0$(respectively β = 0,1).So Eve will have the capacity to recoup around $\frac{4}{9}$ of the coefficients of ϕ_i.

So every one of the extra transmissions $E_2, E_3 ..... E_k$ will empower Eve to decide roughly $\frac{2}{9}$ of the coefficients of ϕ_i, and will likewise give data about another $\frac{4}{9}$ of the coefficients. Subsequently, it takes just a couple of transmissions for Eve to decide each coefficient of ϕ_i , and after that a brute force look on the rest of the coefficients will enable Eve to recoup ϕ_i and the message M.

then perform a brute force assault for finding all possible coefficient with a limited time. so we must not do send a message each time with one public key (P)—

4.1.5 Protection from PQ Attacks

In classical computer we have traditional bits being either 0 or 1 while, in quantum mechanics, we have qubits. Quantum mechanics affirm that a two state framework can be in any superposition of the two premise states.

The best part of Shor's calculation is **Chapter 1**, which is the periodic nd procedure. As an outcome of Shor's algorithm, classical cryptosystems which is based on prime factorization or discrete logarithimic algorithm problem will be shaky under quantum qubit attack. The principle question this raises is the thing that cryptosystems to use in a quantum system. There are much algorithm for a post quantum cryptosystem for example : Merkle'shash-tree PuK signature, McEliece Goppa-code PuKC, and the lattice based CS NTRU and our new algorithm IRTRU.

As a rule, lattices are very hard and the best realized calculations either keep running in exponential time or yields terrible approximations. This is the principle inspiration for lattice based CS. Also, cross section issues are accepted to oppose quantum assaults. Since the disclosure of the factorization quantum calculation by Shor, in 1997, numerous unfruitful endeavors to take care of lattice issues by quantum calculations have been claimed. Thus, it is guessed that there is no quantum calculation that takes care of lattice issues in polynomial time. As an outcome, the IRTRU CS has been sorted as a post-quantum cryptosystem. As saw over, the quantum algorithm part in Shor's calculation utilizes periodic nd strategy. For lattice issues, the primary difculty is that the periodic nd method does not appear to be appropriate. This makes our algorithm as one of the promising CS.

4.2 Environment Setup

- We use a 64-bit computer to implement our algorithm.
- We use C++ language as programming language for coding.
- we test some polynomial in a small range.

Our limitation is the absence of quantum comouter, that's why we can not check in a big range.So it is difficult to tell that what will occur when it will be done in a big bg range suppose in 1000 bits.But we have mathematical proof and con-crete example for it. We use a 64-bit computer

4.3 Complexity Analysis of IRTRU

4.3.1 IRTRU performance

Table 4.2: Hypothetical Operating Specification

Operation Characteristics	Formula
Plain Text Block	$(N-K)Log_2 l$ bits
Encrypted Text Block	$Nlog_2 u$ bits

Table 4.3: RunTime Complexity

Operation Characteristics	Formula
Encryption Speed	$O(N^2)$ Operations
Decryption Speed	$O(N^2)$
Key Generating Speed	$(N-K)log_2 N$

Table 4.4: Comparison with well known public key

Crypto-System	Encryption Speed	Decryption Speed
RSA	$log(N)$	$log(N)$
NTRU	$N\,log(N)$	$n\,log(N)$
AES	$O(N)$	$O(N)$
DES	$O(N)$	$O(N)$
3-DES	$O(N)$	$O(N)$
IRTRU	$(n-k)\,log_2(N)$	$N\,log_2(N)$

4.4 Summary

In these area a portion of the general attack against the IRTRU CS. were talked about, for example, another private keys attack, brute force attack, various message attack with same PuK and lattice section attack (future work). To start with, in substitute private key attack, the elective private keys f^a is utilized to unscramble indistinguishable message from f. Second, we took a gander at the brute force attack and an enhancement, which we called meet in the middle (not in detail form). ultimately, in the various message attack with same PuK, it demonstrates that communicating something specific on numerous occasions with a similar PuK h ought to be maintained a strategic distance from.

Chapter 5

Conclusions

5.1 Summary of Research

In this study a new algorithm for improving efficiency in the time of encryption and decryption and for increasing security of data when passing through network is proposed. It improves the security of the data by increasing confusion and diffusion of the data and cipher-text.If an attacker will try to find out the original data by brute-force attack, mid-in-the middle attack, multiple transmission attack, lattice attack, post quantum attack by a quantum computer, He/She can can break the encryption process, though shor prove that there is no polynomial time algorithm to break a function ,it must be cost exponential time.

5.2 Future Work

The work presented here can be extended to include more research problems to be solved for efficient solutions.

- Analyze Lattice Attack.
- Analyze space and memory complexity of IRTRU.
- Improving security by fixing the co-efficent range of PuK.

- implement digital signature of IRTRU.
- implement data authentication method of IRTRU.

Bibliography

[1] R.L. Rivest, A. Shamir, and L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Commun. ACM, Feb. 1978, 21(2): 120-126. Cham: Springer International Publishing.

[2] McEliece, R. J. and Shearer, J. B., "A Property of Euclid's Algorithm and an Application to Pad Approximation", SIAM J. Appl. Math., Vol. 34, No. 4, June 1978, pp. 611615.

[3] Shor, P. in *Proc. 35th Annu. Symp. on the Foundations of Computer Science* (ed. Goldwasser, S.) 124134 (IEEE Computer Society Press, Los Alamitos, California, 1994).

[4] D. Coppersmith,(May 1994).The Data Encryption Standard (DES) and its strength against attacks.*IBM Journal of Research and Development (Volume: 38 , Issue: 3 , May 1994)* doi: 10.1147/rd.383.0243

[5] D. Coppersmith & D. B. Johnson & S. M. Matyas,(March 1996). (1994). A proposed mode for triple-DES encryption. *IBM Journal of Research and Development (Volume: 40 , Issue: 2 , March 1996)*, doi: 10.1147/rd.402.0253.

[6] V Rijmen & J Daemen,United States National Institute of Standards and Technology (NIST)(2001).Advanced Encryption Standard (AES). *Federal Information Processing Standards Publication 197, Nov. 26 2001*

[7] Darrel Hankerson & Alfred J. Meneze & Scott Vanstone (2003).Guide to Elliptic Curve Cryptography *Springer-Verlag Berlin, Heidelberg ,2003* , ISBN:038795273X

[8] Hoffstein J., Pipher J., Silverman J.H. (1998) NTRU: A ring-based public key cryptosystem. In: Buhler J.P. (eds) *Algorithmic Number Theory. ANTS 1998. Lecture Notes in Computer Science, vol 1423. Springer, Berlin, Heidelberg.* doi: https://doi.org/10.1007/BFb005486

[9] Mikls Ajtai ,IBM Almaden Research Center, San Jose, CA (1998).The shortest vector problem in L2 is NP-hard for randomized reductions (extended abstract). *Proceeding STOC '98 Proceedings of the thirtieth annual ACM symposium on Theory of computing Pages 10-19* .ISBN:0-89791-962-9,doi:10.1145/276698.276705

[10] O. Goldreich, D. Micciancio, S. Safra, J.-P. Seifert, "Approximating shortest lattice vectors is not harder than approximating closest lattice vectors", *Information Processing Letters*, vol. 71, no. 2, pp. 55-61, 1999.

[11] H. Nussbaumer La Gaude Laboratory, IBM France, France."Fast polynomial transform algorithms for digital convolution".*IEEE Transactions on Acoustics, Speech, and Signal Processing (Volume: 28 , Issue: 2 , Apr 1980).*ISSN: 0096-3518,DOI: 10.1109/TASSP.1980.1163372

[12] Henning Vetter, Vishakan Ponnampalam , Magnus Sandell and Peter Adam Hoeher in(2009)."Fixed Complexity LLL Algorithm".*IEEE Transactions on Signal Processing (Volume: 57 , Issue: 4 , April 2009).*DOI: 10.1109/TSP.2008.2011827

List of Notations

CVP - Closest Vector Problem

SVP - Shortest Vector Problem

AES - Advanced Encryption Standard

DES - Data Encryption Standard

3DES - Triple Data Encryption Standard

PQ - Post Quantum Cryptography

PC - Personal **Computer**

ECC - Elliptic Curve Cryptography

Qcomputer - Quantum Computer

PuKC - Public Key Cryptosystem

PvKC - Private Key Cryptosystem

PQC - Post Quantum Cryptography

CSystem - CryptoSystem

EEA - Extended Euclidean Algorithm

EA - Euclidean Algorithm

PvK - Private key

PuK - Public key

Eprocess - Encryption Process

Dprocess - Decryption Process

GVH - Gauss Volume Heuristic

CPR - Convolution Polynomial Ring.