**NAME: RONIT MAHESHWARI**

GROUP NO: 15

**NTRU ALGORITHM:**

**What is NTRU?**

In the year 1996, the first lattice-based cryptosystem was introduced by three researchers namely Jeffrey, Jill and Joseph and this system was named as NTRU. The main agenda of designing this system was public key encryption which was later on extended to the KEM, also known as key encapsulation mechanisms. In the comparison with classical RSA and elliptical curve encryption algorithms, the NTRU is must efficient, scalable and secure. Specifically, for the post-quantum cryptography also, NTRU is essential and aims to make sure quantum attacks are mitigated, avoiding damage to the critical systems (Gökce, Gökce and Cenk, 2024).

**NTRU Design**

The designing for NTRU algorithm has been done based on algebraic operations that are defined in a polynomial ring so that it can work differently than classical algorithms. The main components that are involved in the designing of this algorithm are given below with their mathematical expressions (Kumar, Das and Gangopadhyay, 2024).

* **Polynomial Ring**: The first component is based on forming out a polynomial ring where all operations are performed in it. Also, the mathematical expression that have been used in it is ***Z[x]/(xN−1).***
* **Key Generation:** Another component is key generation where private and public keys are generated differently as explained below.

1. **Private Key**: For the private key, the two polynomials are taken into consideration such as **f(x) and g(x)** where their coefficients are considered to be small set.
2. **Public Key**: For the public key, the polynomial **h(x)** is used and is computed as ***h(x) = g(x) \* f-1(x) mod q***.

* **Encryption**: Once the public and private keys are generated, the plain text polynomial m(x) can be encrypted using the public key polynomial h(x) for creating a random key r(x). The expression for the same is ***e(x) = r(x) ∗ h(x) + m(x) mod q***.
* **Decryption**: After the encryption, the decryption is carried out following the private key polynomial function f(x) which will have expression as ***c(x) = f(x) \* e(x) mod q***.

**Key Sizes and Parameters**

When it comes to security, it can be said that it primarily depends upon three different parameters such as N (Degree of polynomials), q (Modulus), and p (Small Integer). The N is used for governing the dimension of the lattice, the q is a type of integer that defines modulus space and p refers to the modulus for plain text message space. For each different key, there are different levels and sizes as given in the below attached table (Elkabbany et al., 2023).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Security Level | N | q | p | Public key Size | Private Key Size |
| NTRUEncrypt (112-bit) | 509 | 2048 | 3 | 1632 bytes | 1632 bytes |
| NTRUEncrypt (128-bit) | 677 | 2048 | 3 | 2176 bytes | 2176 bytes |
| NTRUEncrypt (256-bit) | 821 | 4096 | 3 | 3280 bytes | 3280 bytes |

Security Mechanisms of NTRU

In terms of security, it can be said that NTRU’s security is defined under the lattice theory which assures that quantum attacks are resistant to both computers. The main security mechanism of NTRU is focusing on lattice-based cryptography and their respective techniques like shortest vector problem, learning with errors and closest vector problem. Also, some other security mechanisms include key generation, hardness of polynomial inversion, use of noise as a blinding factor, and modular arithmetic in polynomial rings (Qin et al., 2021).

## Mechanism 1: Lattice-based Security

The first mechanism for security is lattice-based security where it is also considered to be one of the robust systems when it comes to protecting quantum based attacks. In order to do so, the given below problems are taken into consideration:

* The first problem is based on shortest vector problem where it asks for shortest non-zero vector for computation. When the public key generation process is carried out for creating a lattice, it is mainly done using polynomial multiplication and even modular reduction approaches. Based on these parameters, the higher level of security is attained by the NTSU algorithm (Kumar, Das and Gangopadhyay, 2024).
* Another method is learning with errors (LWE) where a new noise polynomial is introduced abbreviated as r(x) and it is then multiplied with public key and plain text message, referred to as h(x) and m(x). As it creates a noise, it will make sure private key is not recovered by attackers.
* The last method is closet vector problem where the private key received by the receiver will be used for correcting message following closest vector of computation. The decryption will be requiring the solving CVP which is challenging for attackers (Qin et al., 2021).

## Mechanism 2: Key Generation and Hardness of Polynomial Inversion

Another mechanism that can be used for obtaining higher level of security is key generation process where the NTRU must be used for creating out a public-private key pair on the basis of two polynomials like f(x) and g(x). Herein, the f(x) is the private key and its inverse will be -1 for computation of the public key. Furthermore, when the computation of a public key is done, the attacker can only do if there is h(x) and q details else there is no chance for obtaining the message. This is how the key generation and hardness of polynomial inversion ensures security of the selected algorithm’s working (Liang et al., 2024).

## Mechanism 3: Noise (Blinding Factor)

The third mechanism is primarily based on using noise during encryption with the help of random blinding polynomial which is abbreviated as r(x). This works by adding the noise to the cipher text and this will ensure that relationship amongst the message m(x) and cipher text e(x) is obfuscated. This ensures that even if attacker is able to obtain the cipher texts, there are no chances left for extracting information about the plain text from the private key due to the noisy blinding factor (Kadykov, Levina and Voznesensky, 2021).

## Mechanism 4: Modular Arithmetic in Polynomial Rings

The working of NTRU in a polynomial ring is expressed as *Z[x]/(xN−1)* and this is performed on a large integer q so that the combination can be made for arithmetic and the polynomial operations.When it is done, the public key polynomial would be h(x) and the inverse of the key would be f(x0 with polynomial g(x) as modulo q (Liang et al., 2024).

Advantages and Disadvantages

The table given below lists out advantages and disadvantages of NTRU algorithm in the post-quantum work.

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| * NTRU algorithm is quantum resistant which means it may not be attacked by any potential attacks (Zhao and Ye, 2023). * For the speed and reliability also, the NTRU works on polynomial multiplication and modular reduction, improving its speed. * Based on its ability to adjust parameters, the scalability is higher in this algorithm. * The sizes are not small in NTRU but these keys are compact in comparison with RSA algorithm. | * The key sizes for public and private keys are larger which sometimes lead to higher load. * For the parameter sensitivity, there is a need for careful tuning to avoid exposure to attacks. * This algorithm is vulnerable to side channel attacks when power analysis is done (Zhao and Ye, 2023). |

References

Elkabbany, G.F. *et al.* (2023) 'Lightweight computational complexity stepping up the NTRU Post-Quantum algorithm using parallel computing,' *Symmetry*, 16(1), p. 12. https://doi.org/10.3390/sym16010012.

Gökce, N.Y., Gökce, A.B. and Cenk, M. (2024) 'A fast NTRU software implementation based on 5-way TMVP,' *Journal of Information Security and Applications*, 81, p. 103688. https://doi.org/10.1016/j.jisa.2023.103688.

Kadykov, V., Levina, A. and Voznesensky, A. (2021) 'Homomorphic Encryption within Lattice-Based Encryption System,' *Procedia Computer Science*, 186, pp. 309–315. https://doi.org/10.1016/j.procs.2021.04.149.

Kumar, V., Das, R. and Gangopadhyay, A.K. (2024) 'GR-NTRU: Dihedral group over ring of Eisenstein integers,' *Journal of Information Security and Applications*, 83, p. 103795. https://doi.org/10.1016/j.jisa.2024.103795.

Liang, Z. *et al.* (2024) 'Compact and efficient KEMs over NTRU lattices,' *Computer Standards & Interfaces*, 89, p. 103828. https://doi.org/10.1016/j.csi.2023.103828.

Qin, Z. *et al.* (2021) 'A compact full hardware implementation of PQC algorithm NTRU,' *A Compact Full Hardware Implementation of PQC Algorithm NTRU* [Preprint]. https://doi.org/10.1109/cisce52179.2021.9446042.

Zhao, Z. and Ye, Q. (2023) 'Revisiting lower dimension lattice attacks on NTRU,' *Information Processing Letters*, 181, p. 106353. https://doi.org/10.1016/j.ipl.2022.106353.