JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 709 - 716
https://publishoa.com
ISSN: 1309-3452

Novel of NTRU public key generate by QKD

Saba Alaa Abdulwahhab[1], Awni M. Gaftan[2], Qasim Mohammed Hussien[3]
Saba.programmer12.tu.edu.iq [1]
Awny.muhammed@tu.edu.iq[2]
kasimalshamry@tu.edu.iq[3]
[1,3]Department of Computer, College of Computer Science and Mathematics, Tikrit, Iraq
[2]Department of Financial and Banking Sciences, College of Administration and Economics, Tikrit, Iraq

ABSTRACT
Most of the data has become widely transmitted in digital devices, which affects the security of this data due to a large number of hackers, and to protect it, encryption is used as one of the reliable methods to ensure data security. In this paper, we devised novel method for generating the public key of NTRU algorithm which is classify as a post quantum cryptography in third round of NIST, the generation properties dependent of quantum mechanics using QKD algorithm, where a set of random numbers was generated, which were chosen randomly in the generation of the key, in this case, the inverse of polynomial was dispensed, which was one of the disadvantages of NTRU algorithm.
Keyword: post quantum, quantum key distribution, NTRU, public key, cryptography.

1. INTRUDUCTION

When a quantum computer becomes accessible, Shor's algorithm will pose a security threat to all classical cryptography protocols. Following Shor's algorithm and Grover's search method, research in quantum computing increased, even there are currently no known mathematical shortcuts to these methods[1], which means that every feasible combination (or brute force) must be evaluated to obtain the key number that will unlock the algorithm [2]. So Information in a quantum system cannot be copied (Nocloning Theorem) or read by an unauthorized party. As a result without quantum-resistance encryption, confidence in information systems that manage vital information will be unattainable shortly [3].

Quantum cryptography is a field that uses quantum physics to secure the security, confidentiality, and reliability of a system [4]. Quantum computing is a new way to find answers to some computer problems, like the factorization problem and the discrete logarithm problem, which is a lot faster than traditional computing. With this new method, quantum computers can find these answers in polynomial time [1]. In the area of quantum cryptography, QKD is a new technology. Unlike traditional encryption algorithms, which rely on mathematical complexity as a security strength criterion [5], post-quantum cryptography is a type of security that is resistant to both classical and quantum attacks, if it operates effectively on classical computers but will withstand quantum attacks [6, 7].

In this paper we propose novel method of generation public key of NTRU algorithm using QKD, when quantum mechanics is programmed to generate impenetrable random numbers that are quantum-resistant, according NIST , where post quantum is resistance to quantum attack , so this method in addition to increasing safety, eliminated the inverse of polynomial , which is one of the disadvantages of NTRU algorithm.

2. Methodology

In this section explain the detail of how generation the key using quantum key distribution then encryption and decryption using NTRU algorithm on plain text save in the file to test the randomness NIST testing.

2.1 Quantum key distribution (QKD)

The necessity for unconditionally secure communication protocols gave rise to quantum cryptography and quantum key distribution. Classical cryptography focuses on the computational complexity of mathematical problems for security[8]. Factoring a big number into primes is one example and the basic security assumption of RSA. Classical cryptography systems are endangered by Shor's factoring techniques. Shor's approach is only just beginning to be implemented on a quantum computer in practice, so a QKD-based protection mechanism must be developed before a quantum computer can breach current encryption systems [9]

In the case study of the using QKD we proposed Eavesdropping attacks to eliminite the eav intercepted or modifiy the data between Alice and Bob .

2.2 STEP OF GENERATION

In this paper proposeAlice generates a qubit then transmits it to Bob, then measures it in the X-basis , propose if Eve intercepts and reads it; Bob then measures it in X-basis, Alice creates random bits It can be any number specifyby Alice , Eav try Interception , Suppose Bob and Alice used the same basis, then added it to the list of 'good' bits, and both identified which basis they used for each qubit. If Bob measures a qubit the same way Alice did, utilise it for their shared secret key, otherwise the rejection is done. Finally, they exchange a random sample of their keys, and if the samples match, they may be

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 709 - 716
https://publishoa.com
ISSN: 1309-3452

certain that their transmission was successful (within a tiny margin of error). Her converts to decimal using summation in this step, and the final result is 4494, 4671.
This model was created to anticipate the result of the NTRU public key by random choosing, which is discussed in the next section.

2.3 NTRU Algorithm

The NTRU (Number Theory Research Unit) is a collection of mathematical algorithms for manipulating lists of very small integers and polynomials[10]. NTRU operations are based on objects in the $R = \frac{Z[X]}{X^N-1}$truncated polynomial ring As a consequence[11], NTRU may achieve high speeds while using little computing resources [12]. The NTRU key generation technique requires computing the modular multiplicative inverse of F modulo p and q [13], making it the first secure public-key cryptosystem that does not depend on factorization or discrete logarithm concerns[14]. Let f, g polynomial of the form

$f = a_0 + a_1X + a_2X0^1 + a_3X1^3$+……..+$a_{n-1}X^{N-1}$+$a_nX^N$(1)

$g = a_0 + a_1X + a_2X0^1 + a_3X1^3$+……..+$a_{n-1}X^{N-1}$+$a_nX^N$(2)

1. NTRU Key generation

To make public and private keys, we first need to find the multiplicative inverse of $f\ mod\ p$ and $g\ mod\ q$ so that the public and private keys match [12, 15]

$$h = F_q \star g\ mod\ \text{q}$$

_ This step is eliminate in this paper and then generate the public key H by using QKD

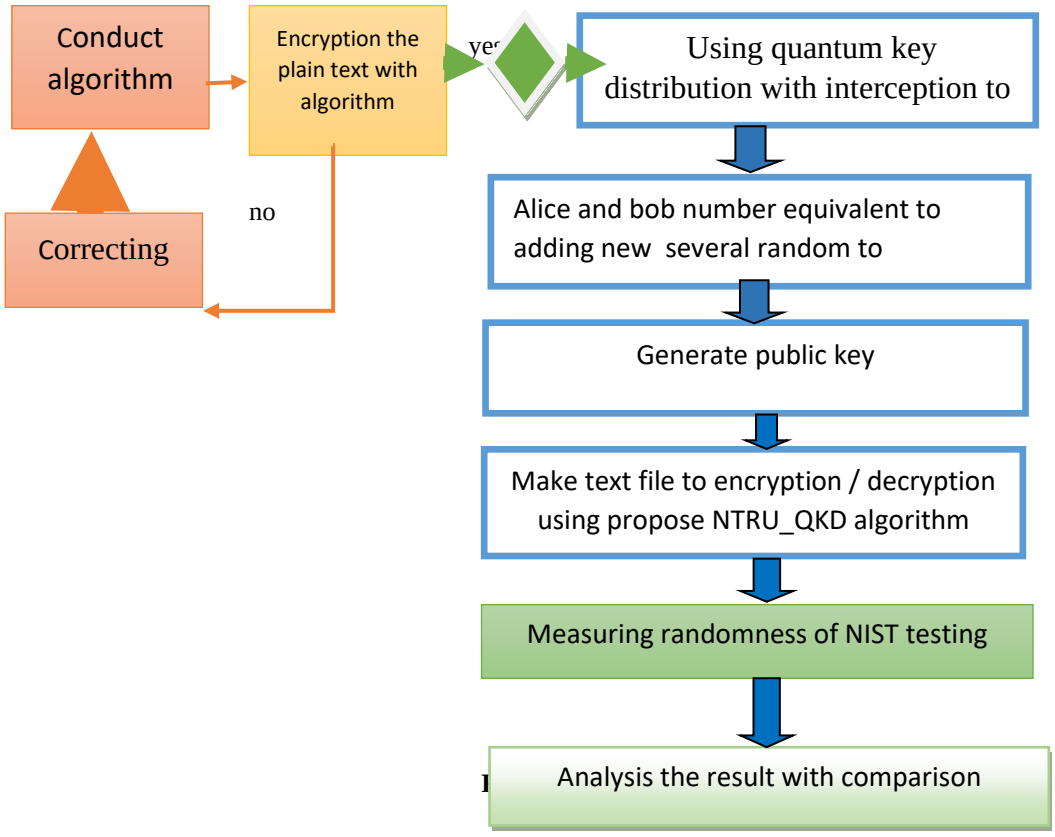
2.ENCRYPTION OF NTRU

Bob can transmit an encrypted message to Alice using the NTRU equation and a public key [16]

$e = \text{rh} + \text{m}(\text{mod q})$ (5)

Bob's encrypted message now. Bob may now transmit e to Alicen as equation (5)

2.4 flowchart



JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 709 - 716
https://publishoa.com
ISSN: 1309-3452

3. Decryption of NTRU

Alice wants to decrypt Bob's message now that she has it. Alice now computes the polynomial defined by [16-18].

$a = f_e (\text{mod q})$ (6)

She then computes the polynomial b defined by the expression:

$$\mathbf{b} = \mathbf{a}(\text{mod } \mathbf{p}) \quad (7)$$

Finally, Alice computes the polynomial **C** defined by the expression:

$$\mathbf{c} = \boldsymbol{f}_p \mathbf{b} \text{ (mod p)} \quad (8)$$

Bob's original message m will be represented by this polynomial **C.**

4. Result and discussion

this section explain creating the fully using quantum key distribution ,most of Previous studies discussed the back draw of inverse polynomial in the complex step ,to generate the inverse and lost time when generation in this section . This complicated step will be eliminated and replaced with a faster step by using the physical properties of quantum mechanics, in this study choosing QKD with simple modification to get compatibility with the NTRU algorithm as example below

```
==== NTRU_QKD generates public key =====
Values used:
 N= 128
 p= 5
 q= 71
========

Bob picks two polynomials (g and f):
f(x)=  [-1, 1, 1, 0, -1, 0, 1, 0, 0, 1, -1]
g(x)=  [-1, 0, 1, 1, 0, 1, 0, 0, -1, 0, -1]

====Now compute F_p  ===
F_p: [0, 1, 1, 3, 4, 1, 2, 4, 3, 0, 1, 3, 0, 3, 2, 3, 3, 0, 3, 4, 3, 2, 3, 0, 4, 1, 2, 3, 0, 1, 2, 2, 3, 1, 4, 0, 4,

====And finally h====
H (NTRU_QKD Public Key):  [47, 55, 60, 24, 55, 11, 20, 10, 48, 16, 12, 4]

====Let's Encrypt====
Alice's Message:         [1, 0, 1, 0, 1, 1, 1, 1]
Random:                  [-1, -1, 1, 1, 1]
Encrypted message:       [50, 58, 16, 19, 61, 11, 44, 17, 69, 27, 37, 6, 5, 18, 9, 20]
--- 0.0007798671722412109 seconds ---
Converted binary list is :  ['110010', '111010', '10000', '10011', '111101', '1011', '101100', '10001', '1000101', '

====Let's decrypt====
Decrypted message:       [1, 0, 1, 0, 1, 1, 1, 1]
--- 0.0024039745330810547 seconds ---
```

Figure 2: the result of encryption decryption of NTRU_QKD

By choosing a random number generate by QKD using IBM lab h=[47,55,60,24,55,11,20,10,48,16,12,4] , we can generate any random number by determine the N , and the rang of number in 2^n , so the key is not static it can change if any interception or if Alice change manually.

4.1 Randomness testing of NTRU_QKD result

In this section after conversion to 1, 0 to calculate the randomness result below in Table 1 , Table 2 , Table 3 .

Table 1. NIST randomness result

N.	Type of NIST test	P-Value	Conclusion
1	**Frequency Test (Monobit)**	0.12282264810139258	Random

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 709 - 716
https://publishoa.com
ISSN: 1309-3452

2	**Frequency Test within a Block**	0.12282264810139258	Random
3	**Run Test**	0.9503121312132852	Random
4	**Longest Run of Ones in a Block**	0.0	Non-Random
5	**Binary Matrix Rank Test**	-1.0	Non-Random
6	**Discrete Fourier Transform (Spectral) Test**	0.9435575426376712	Random
7	**Non-Overlapping Template Matching Test**	0.9999999981564897	Random
8	**Overlapping Template Matching Test**	nan	Non-Random
9	**Maurer's Universal Statistical test**	-1.0	Non-Random
10	**Linear Complexity Test**	-1.0	Non-Random
11	**Serial test:**	0.4989610874592239	Random
12	**Approximate Entropy Test**	1.0	Random
13	**Cumulative Sums (Forward) Test**	1.0153767805840006	Random
14	**Cumulative Sums (Reverse) Test**	0.24563978734012576	Random

Table 2. NIST randomness result

N	**Type of NIST**	**State**	**Chi Squared**	**P-Value**	**Conclusion**
15	**Random Excursions Test**	-4	0.2857142857142857	0.9979031595107858	Random
		-3	0.4	0.9953295932358704	Random
		-2	0.6666666666666666	0.984747879018509	Random
		-1	2.0	0.8491450360846096	Random

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 709 - 716
https://publishoa.com
ISSN: 1309-3452

		+1	14.0	0.01560941610026691	Random
		+2	16.666666666666664	0.005177344251965914	Non-Random
		+3	16.599999999999998	0.005324337409375595	Non-Random
		+4	30.571428571428573	1.138085401980086e-05	Non-Random

Table 3 NIST randomness result

N	**Type of NIST**	**State**	**COUNTS**	**P-Value**	**Conclusion**
16	**Random Excursions Variant Test**	+1.0	4	0.31731050786291415	Random
		+2.0	3	0.7728299926844475	Random
		+3.0	1	0.8230632737581215	Random
		+4.0	1	0.8501067391385259	Random
		+5.0	1	0.8676323347781927	Random

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 709 - 716
https://publishoa.com
ISSN: 1309-3452

		+6.0	1	0.8801684549067255	Random
		+7.0	1	0.8897069355331012	Random
		+8.0	1	0.897278961260083	Random
		+9.0	4	0.8083651559145103	Random

3.2 Comparison of NIST result between NTRU_QD and NTRU_QKD

The chart below show the result of NIST testing between NTRU_QD and NTRU_QKD so the best result should be between the rang 0 to 1 as the figure 2. Show the NTRU_QKD in red color is best one randomness testing should be between o and 1 is best result .

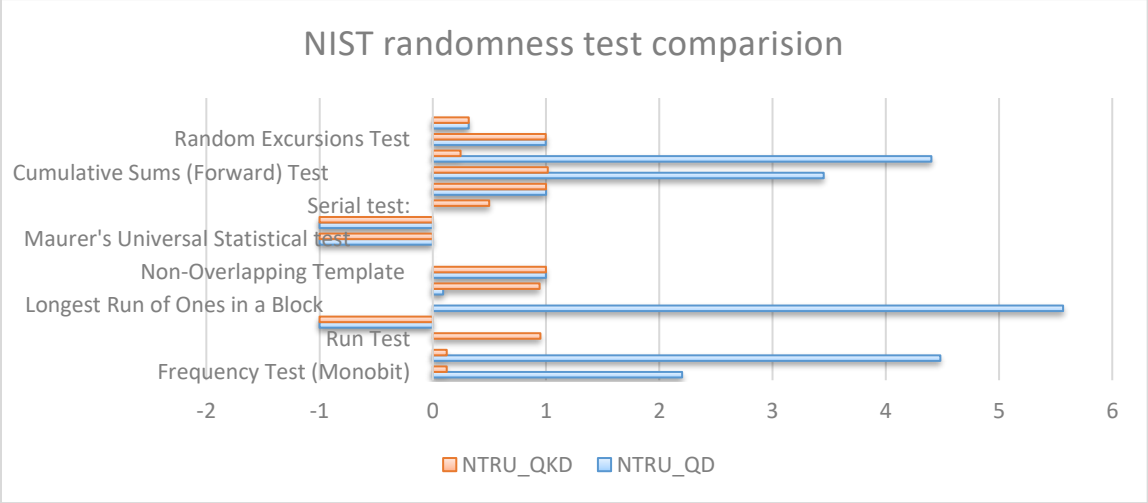


Figure 3.The comparison NIST result

3.3 An Analysis of Performance

This section compares the performance of these approaches with the original NTRU cryptosystem. First suggestion uses four prime parameters (N, QK, p, q). Original NTRU parameters are three (N, p, q). Original NTRU and suggested techniques have the same plaintext block size. Complexity of proposals equals original NTRU:

Table 4 the complexity of NTRU

Characteristics	**The complexity[19]**
Plaintext Block	N $\log_2$ p bits
Encrypted Text Block	N $\log_2$ q bits
Speed of Encryption	$O(N^2)$ operations
Speed of Decryption	$O(N^2)$ operations

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 709 - 716
https://publishoa.com
ISSN: 1309-3452

So, the modification for the both two proposal can generate approximately N versions of key sequence with the rang 2^N from the original public key, this means that we do not repeat the public key with each block, but each block is encrypted with a key that is completely different from the other keys that precedes it and there is no relationship between them that may be exploited by the attacker to break these keys.

3.4 Analysis of the Encryption and Decryption Times

The proposals and the original NTRU cryptosystem both run on the same laptop with the same parameters (N=128, p=5, q=71), which results in the following encryption and decryption times in second:

Table 5 Time execution of NTRU

Algorithm	Encryption	Decryption
NTRU	**0.00253**	**0.00306**
NTRU_QD	0.0156	0.0312
NTRU_QKD	**0.0007**	**0.0024**

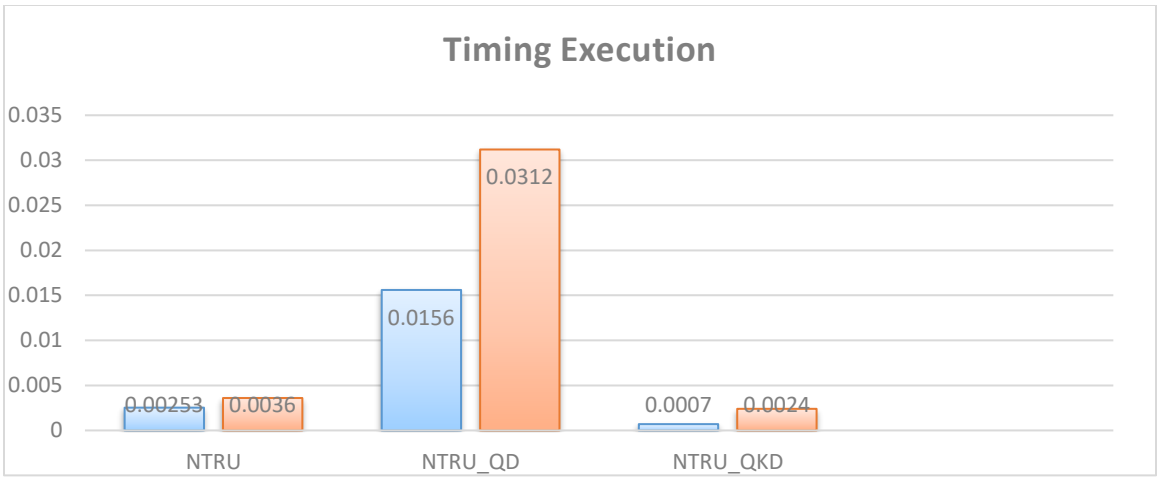


Figure 4. Execution time of NTRU

As figure 3 shows the NTRU_QKD is best in time execution of encryption / decryption from NTRU_QK or original NTRU.

5. CONCLUSION

Since the NTRU algorithm has been around for a long time, hackers would find it easier to find a way to get around it and steal money, liquid assets, or private information from online accounts. The proposed method increases the complexity of breaching into such accounts by introducing additional layers of security that are purely based on fundamental mathematical and physical properties using quantum key distribution while ensuring a practically feasible time for execution. Through the overview of the topic, also the paper compression between two NTRU modification and result shows the last one (NTRU_QKD) is best in randomness NIST result , and best in execution time , we can recommend future work on choosing a specific attack method to test whether if this system can be broken by quantum attack or even the classical methods used previously.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 709 - 716
https://publishoa.com
ISSN: 1309-3452

REFRENCES

[1] A. Kumar and S. Garhwal, "State-of-the-Art Survey of Quantum Cryptography," *Archives of Computational Methods in Engineering*, pp. 1-38, 2021.

[2] L. O'Connor, C. Dukatz, L. DiValentin, and N. Farhady, "Cryptography in a post-quantum world: preparing intelligent enterprises now for a secure future. Accenture Labs," ed, 2018.

[3] K. Isirova and O. Potii, "Requirements and Security Models for Post-Quantum Cryptography Analysis," in *Proceedings of the PhD Symposium at 13th International Conference on ICT in Education, Research, and Industrial Applications co-located with 13th International Conference on ICT in Education, Research, and Industrial Applications (ICTERI 2017)*, 2017, pp. 36-41.

[4] D. Alvarez and Y. Kim, "Survey of the Development of Quantum Cryptography and Its Applications," in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, 2021, pp. 1074-1080.

[5] A. Broadbent and C. Schaffner, "Quantum cryptography beyond quantum key distribution," *Designs, Codes and Cryptography*, vol. 78, pp. 351-382, 2016.

[6] F. Song, "A note on quantum security for post-quantum cryptography," in *International Workshop on Post-Quantum Cryptography*, 2014, pp. 246-265.

[7] M. Kumar and P. Pattnaik, "Post Quantum Cryptography (PQC)-An overview," in *2020 IEEE High Performance Extreme Computing Conference (HPEC)*, 2020, pp. 1-9.

[8] S. S. Gill, A. Kumar, H. Singh, M. Singh, K. Kaur, M. Usman, *et al.*, "Quantum computing: A taxonomy, systematic review and future directions," *Software: Practice and Experience*, vol. 52, pp. 66-114, 2022.

[9] A. I. Nurhadi and N. R. Syambas, "Quantum key distribution (QKD) protocols: A survey," in *2018 4th International Conference on Wireless and Telematics (ICWT)*, 2018, pp. 1-5.

[10] U. M. Awnon Bhowmik, "Enhancing the NTRU Cryptosystem," *International Journal of Computer Applications*, vol. 179, 2020.

[11] M. Albrecht and L. Ducas, "Lattice Attacks on NTRU and LWE: A History of Refinements," *Cryptology ePrint Archive*, 2021.

[12] G. J. Nyokabi, M. Salleh, and I. Mohamad, "NTRU inverse polynomial algorithm based on circulant matrices using gauss-jordan elimination," in *2017 6th ICT International Student Project Conference (ICT-ISPC)*, 2017, pp. 1-5.

[13] J. N. Gaithuru, M. Salleh, and I. Mohamad, "NTRU inverse polynomial algorithm based on the LU decomposition method of matrix inversion," in *2017 IEEE Conference on Application, Information and Network Security (AINS)*, 2017, pp. 1-6.

[14] Z. Qin, R. Tong, X. Wu, G. Bai, L. Wu, and L. Su, "A Compact Full Hardware Implementation of PQC Algorithm NTRU," in *2021 International Conference on Communications, Information System and Computer Engineering (CISCE)*, 2021, pp. 792-797.

[15] M. Bufalo, D. Bufalo, and G. Orlando, "A Note on the Computation of the Modular Inverse for Cryptography," *Axioms*, vol. 10, p. 116, 2021.

[16] B. Clark, "Understanding the NTRU Cryptosystem," 2019.

[17] A. Pellet-Mary and D. Stehlé, "On the hardness of the NTRU problem," in *International Conference on the Theory and Application of Cryptology and Information Security*, 2021, pp. 3-35.

[18] R. Asif, "Post-quantum cryptosystems for Internet-of-Things: a survey on lattice-based algorithms," *IoT*, vol. 2, pp. 71-91, 2021.

[19] R. Ebrahimi Atani, S. Ebrahimi Atani, and A. Hassani Karbasi, "NETRU: A non-commutative and secure variant of CTRU cryptosystem," *The ISC International Journal of Information Security*, vol. 10, pp. 45-53, 2018.