quantum attacks. Furthermore, AES suffers from key exchange problem which are limitations.

The article in [20], posited that the efficiency of ECC depended on point multiplication and the lattice multiplication operation could be applied to it, suggesting that ECC is efficient but most suitable for environments where keys of small size could be applied. They further stated that wireless sensor networks is the best area where ECC is to be applied, which enhances wireless devices to perform end to end secure communication efficiently. They presented methods that could be used for lattice multiplication operation and proposed the use of binary method in Lattice multiplication suggesting that it promotes the speed and accuracy of the multiplication. They presented simulation results that validated the proposed method and analysis. Despite the fastness in execution of the binary method in lattice multiplication, the limitation of the proposed scheme is that ECC increases the size of encrypted data. Additionally, the ECC algorithm is complex to deploy, increasing the chances of implementation errors thus this affects the security of the algorithm.

The research by [30] revealed that a lot of post quantum solutions are been developed and submitted to National Institute of Standards and Technology (NIST) for onward standardisation and possible deployment. The study also implemented NTRU cryptosystem on an embedded system. Using a python driven development framework for the design. evaluation was carried out based on speed and consumption of resources metrics. The experiment carried out revealed that operations using the python+C programming enhanced performance ranging from 130 to 450 depending on the scenario in the application of the algorithm.

The study by [37] portrayed a technique for data security in the cloud. The authors also presented various techniques and characteristics for big data cloud computing and stated some of the challenges of data security. The authors presented a virtualization technique for safeguarding data in the cloud. However, the study did not simulate the proposed technique and make comparison with similar techniques. Thus, its level of efficiency could be established.

3. The Proposed System Design

The proposed NTRU cryptosystem was applied to the data that will be stored in the cloud. Below is a conceptual architecture for the security of data stored in the cloud.

Quantum attack-resistant crypto schemes such as those that are based on code, hash, multivariate quadratic polynomial or even the lattice (NTRU) cryptosystems are tipped to become alternatives to RSA and ECC [21-24]. Thus, this current study seeks establish if NTRU will be faster in execution when compared to other cryptosystems [25] and [26]. NTRU possess lower complexity and smaller key size which makes it a good alternative for modern cryptography, hence adoptable to computing (Classical and Quantum Computing).

The following variables are used in processing the encryption/decryption of the NTRU cryptosystem;

N - the polynomials in the ring *R* with degree *N-1*.

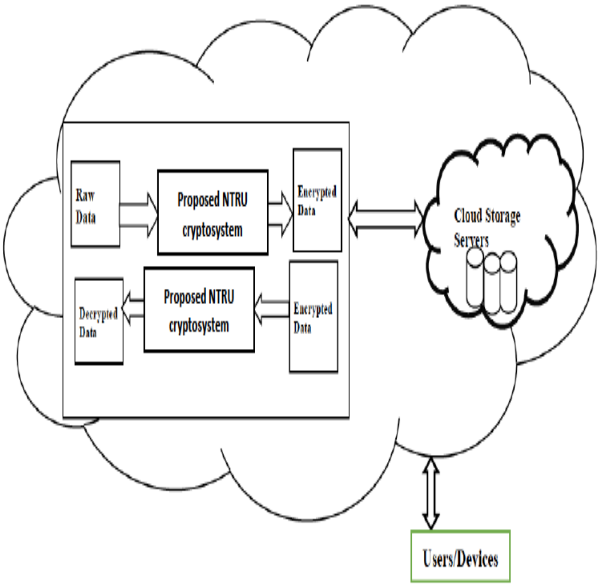

Figure 1. Conceptual Architecture of the proposed cloud data security framework

q - the large modulus, which is used for the reduction of coefficients.

p - the small modulus, which is used for the reduction of coefficients.

f - a component of the private key, which is represented in polynomial form.

g - a polynomial used to process the public key h from f

h - is a polynomial

r - a random blinding polynomial used to distort data

m- is the message to be encrypted represented in polynomial form.

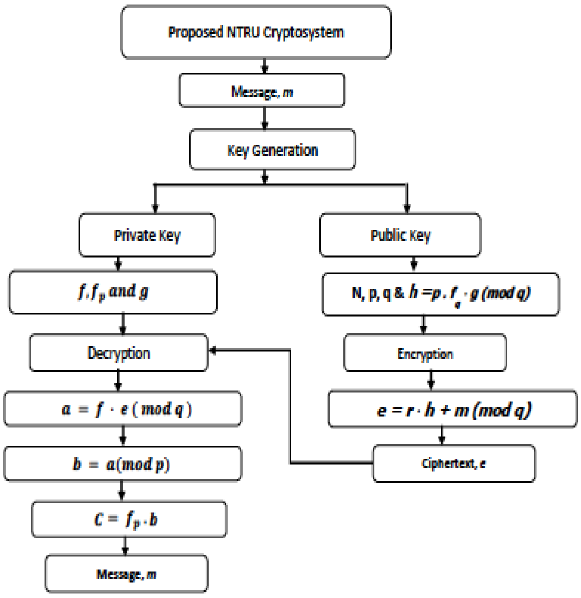

Figure 2. The flow in the proposed Lattice-based NTRU cloud data security system

a. Key Generation in the proposed system

The sender computes $f \cdot f_p = 1(mod\ p)$ and $f \cdot f_q = 1(mod\ q)$ and then processes the public key h using:

$$h = p \,.\, f_q .g\ (mod\ q) \qquad (1)$$

The above equation 1 is computed using lattice multiplication

b. Data Encryption in the proposed system

To encrypt a message, the sender following is processed:

$$e = r \,.\, h + m\ (mod\ q) \qquad (2)$$

c. Data Decryption in the proposed system

The following is computed to decrypt the message

$$a = f \,.\, e\ (moq\ q) \qquad (3)$$

$$b = a\ (mop\ p) \qquad (4)$$

$$C = f_p \,.\, b \qquad (5)$$

The above equation 1, 2, 3 4 and 5 is computed using lattice multiplication.

3.1. Implementation of the Proposed System

To verify the efficiency of the suggested algorithm, symmetric and asymmetric cryptosystems were chosen. Various data sizes were simulated against four algorithms (RSA, ECC, AES, and NTRU), as well as the proposed algorithm, with the sole purpose of determining the throughput of the encryption and decryption execution time of the data used and determining whether the proposed algorithm is a better way of safeguarding cloud data.

On a MacOS computer with an Intel Core i5 processor, 8GB of RAM, and 250GB of hard disk space, the simulation was run with MATLAB.

Table 1. Time Taken for The Generation of Private Key

Input File Size (KB)	ECC(s)	AES(s)	RSA(s)	Existing NTRU(s)	Proposed NTRU(s)
20	0.000191	0.000109	0.00000048	6.1037757	7.1810455
77	0.000193	0.000097	0.00000095	6.2219977	6.8497849
153	0.000189	0.000100	0.00000072	6.2695474	6.9003048
283	0.000185	0.000094	0.00000072	6.1897840	6.8672769
305	0.000180	0.000114	0.00000072	6.2266142	6.8146033
Average time (s)	**0.000188**	**0.000103**	**0.00000072**	**6.202344**	**6.9226031**

Table 2. Time Taken for The Generation of Public Key

Input File Size (KB)	ECC(s)	AES(s)	RSA(s)	Existing NTRU (s)	Proposed NTRU (s)
20	0.000220	0.000109	0.000002	6.103776	7.181046
77	0.000222	0.000097	0.000002	6.221998	6.849785
153	0.000219	0.000100	0.000003	6.269547	6.900305
283	0.000214	0.000094	0.000002	6.189784	6.867277
305	0.000209	0.000114	0.000002	6.226614	6.814603
Average time (s)	**0.000217**	**0.000103**	**0.0000023**	**6.202344**	**6.922603**

Table 3. Time Taken for The Encryption Process of Various Cryptosystems

Input File Size (KB)	ECC (s)	AES (s)	RSA (s)	Existing NTRU (s)	Proposed NTRU (s)
20	0.0097	0.3122	0.0753	320.8390	564.1053
77	0.0080	0.2490	0.7298	1319.4852	2339.64560
153	0.0119	0.2537	0.5514	2616.9852	7724.5771
283	0.0228	0.2471	0.0228	4898.3393	11570.9521
305	0.0181	0.2490	0.7932	5268.4621	37354.0916
Average time (s)	**0.0141**	**0.2622**	**0.4345**	**2884.8222**	**11910.6744**

Table 4. Time Taken for the Decryption Process of Various Cryptosystems

File Size(KB)	ECC(s)	AES(s)	RSA(s)	Existing NTRU(s)	Proposed NTRU(s)
20	0.0032	0.2415	0.0820	477.9566	1446.2688
77	0.0052	0.2383	0.4338	1911.9348	5822.7922
153	0.0096	0.2506	0.7058	3931.4452	11834.1894
283	0.0161	0.2514	1.4746	7313.5120	32812.9468
305	0.0167	0.2780	1.5585	32157.8710	60374.1992
Average time (s)	**0.0102**	**0.2520**	**0.8509**	**9158.5439**	**22458.0793**

Table 5. Total Average Execution Time of Various Algorithms

	ECC	AES	RSA	Existing NTRU	Proposed NTRU
Private Key Execution Time	0.0002	0.0001	0.0000	6.2023	6.9226
Public Key Execution Time	0.0002	0.0001	0.0000	6.2023	6.9226
Encryption Execution time	0.0141	0.2622	0.4345	2884.822	11910.6744
Decryption Execution Time	0.0102	0.2520	0.8509	9158.5439	22458.0793
Total Execution Time	0.0247	0.5144	1.2854	12055.7708	34382.5989
Throughput (KB/S)	33919.0879	1629.1607	651.9196	0.0695	0.02435

The deductions from the simulation carried out are arranged based on the time taken to generate private key, public key, encryption and decryption.

Copyright © 2022, Infonomics Society | DOI: 10.20533/ijisr.2042.4639.2022.0119

a. Private Key Generation time

Figure 3 shows the average time it takes to generate the private key for ECC, RSA, AES, existing NTRU and Proposed NTRU.

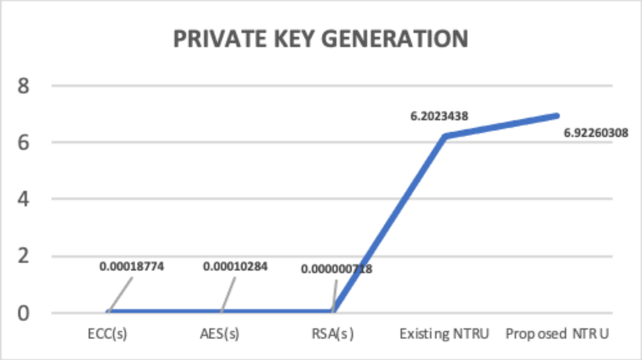

Figure 3. Average Private Key Generation

Equally, regards to NTRU algorithms, it can be deduced from the foregoing that the proposed NTRU cryptosystem takes more time to generate the private key while existing NTRU cryptosystem takes lesser time, this could be as a result of the lattice arithmetic approach, which was introduced in the computation proposed in the NTRU algorithm or the hardware used for the simulation.

b. Public Key Generation time

The Figure below shows the average time it takes to generate the public key for ECC, RSA, AES, exiting NTRU and Proposed NTRU.

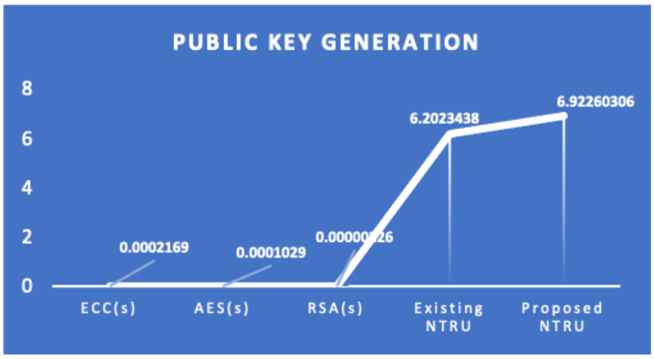

Figure 4. Average Public Key Generation

Similarly, as regards to NTRU algorithms, it can be inferred from the above Figure above that the existing NTRU cryptosystem takes lesser time to generate the key as against the proposed NTRU cryptosystem which takes more time. The lesser time that the proposed NTRU takes could be as a result of introduction of lattice arithmetic that this study introduced to the processing of NTRU as against the polynomial arithmetic, which the existing NTRU algorithm dwells on.

c. Encryption Time

Figure 5 below depicts the average encryption time for ECC, RSA, AES, existing NTRU and Proposed NTRU.

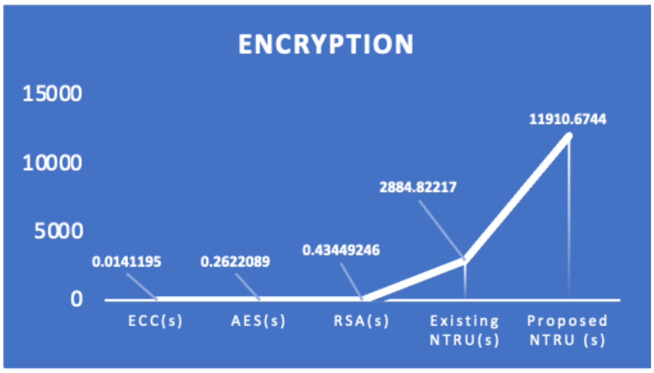

Figure 5. Average Encryption time

With respect to NTRU algorithm, it can be inferred from Figure 5, that the proposed NTRU cryptosystem takes more time to encrypt while the existing NTRU cryptosystem takes lesser time. The more time that the proposed NTRU takes could be as a result of introduction of lattice arithmetic that drives the encryption process. Thus, the proposed NTRU takes more time to encrypt data.

d. Decryption Time

The Figure below depicts the average decryption time for ECC, RSA, AES, existing NTRU and Proposed NTRU.

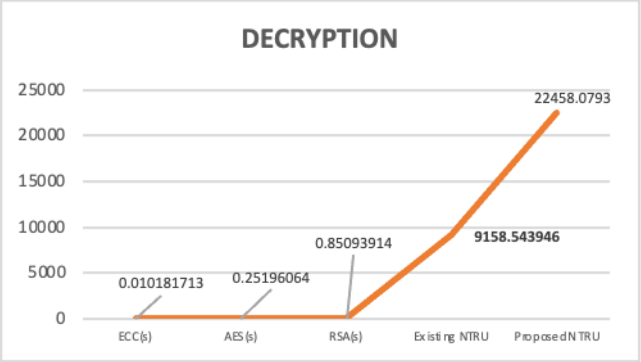

Figure 6. Average Decryption time Generation

In respect to NTRU algorithms, it can be deduced from the above Figure that the proposed NTRU cryptosystem takes more time to decrypt while the existing NTRU cryptosystem takes lesser time. The more time that the proposed NTRU takes could be as a result of introduction of lattice arithmetic that drives the decryption process.

e. Throughput of the Algorithms

The throughput is computed based on the private and public key computation; and also, encryption and decryption execution times of the algorithms.

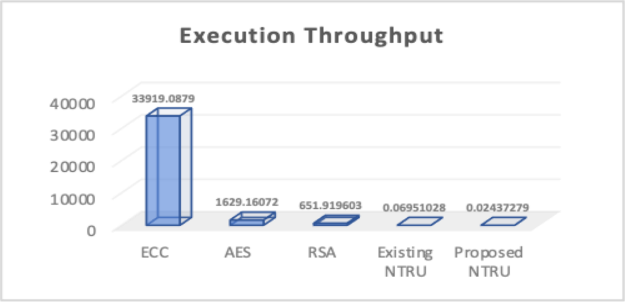

Figure 7. Execution throughput

Figure 7 above shows that ECC has the best throughput, however, the existing NTRU algorithm has a better throughput when compared with the proposed NTRU algorithm.

f. Power consumption of various cryptosystems

If the throughput is calculated correctly, the higher the throughput of a cryptosystem's time complexity, the lower the power consumption [27-29]. As a result, execution throughput is proportional to power consumption. From the Figure 7 above, it can be inferred that ECC has the lowest power consumption. The existing NTRU, on the other hand, consumes less power than the proposed NTRU.

4. Complexity of the Algorithm

Algorithmic complexity is a measure of how long it would take an algorithm to complete a given n-dimensional input. Even with huge values of *n*, a scaling method should compute the result within a finite and reasonable time bound. As a result, as *n* approaches infinity, complexity is estimated asymptotically. While complexity is normally measured in terms of time, it can also be measured in terms of space, which corresponds to the memory requirements of an algorithm. When comparing algorithms or looking for improvements, it's useful to look at their complexity. Computational complexity theory is an area of theoretical computer science that deals with algorithmic complexity. It's vital to note that the paper is only interested in the time complexity order of an algorithm.

a. Time complexity for the Proposed NTRU algorithm

For the computation of the time complexity for the key generation of the proposed NTRU, the computations in the algorithm is considered which is shown below.

i. Complexity of the proposed NTRU Key Generation process

Complexity for modulus arithmetic is $O(n^{1/2})$

Process 1---Time Complexity for modulus lattice multiplication = $O(n^3)$

Process 2--- Time Complexity for modulus lattice multiplication = $O(n^3)$

Process 3--- Time Complexity for modulus lattice multiplication = $O(n^3)$

Process 4 -- Process 4 -- Time Complexity of retuning the output = $O(n)$

Hence, the time complexity for the key generation considering the highest complexity is $O(n^3)$.

ii. Complexity of the proposed NTRU Encryption process

Proposed NTRU - Encryption

```
Input: Parameters for encryption $(m, r, h, q)$
Output: Cipher Text $(e)$
Begin
    i.  Compute $e = r \cdot h + m \ (mod\ q)$
    ii. Return $(e)$
End
```

Complexity for modulus arithmetic is $O(n^{1/2})$

Process 1 -- Time Complexity for computing the modulus lattice multiplication = $O(n^3)$

Process 2 -- Time Complexity of retuning the output = $O(n)$

Hence, the time complexity for the encryption considering the highest complexity is $O(n^3)$.

iii. Complexity of the proposed NTRU Decryption process

Proposed NTRU- Decryption

```
Input: Parameters for encryption $(e, f, p, q)$
Output: Plain Text $(c)$
Begin
    i.   Compute $a = f \cdot e\ (mod\ q)$
    ii.  Compute $b = a(mod\ p)$
    iii. $C = f_p \, . \, b$
    iv.  Return (c)
End
```

Complexity for modulus arithmetic is $O(n^{1/2})$

Process 1: Time Complexity for modulus lattice multiplication = $O(n^3)$

Process 2: Time Complexity for modulus lattice multiplication = $O(n^3)$

Process 3: Time Complexity for lattice arithmetic = $O(n^2)$

Process 4: Process 4 -- Time Complexity of retuning the output = $O(n)$. Hence, the time complexity for the Decryption considering the highest complexity is $O(n^3)$.

Finally, time Complexity of the Proposed NTRU Algorithm =

Time complexity for (Key generation + encryption + decryption) = $O(n^3) + O(n^3) + O(n^3) = 3O(n^3)$. Upon eliminating constants, the time complexity of the proposed NTRU algorithm is $O(n^3)$.

Proposed NTRU- Key Generation

```
Input: Parameters for encryption $(p, f, g, q)$
Output: Keys $(h)$
Begin
    i.   Compute $f \cdot f_p = 1(mod\ p)$ and
    ii.  $f \cdot f_q = 1(mod\ q)$
    iii. $h = p \, . \, f_q \cdot g\ (mod\ q)$
 i.      Return $(h)$
End
```

Copyright © 2022, Infonomics Society | DOI: 10.20533/ijisr.2042.4639.2022.0119

Zalekian et al. in [26], opines that the NTRU algorithm requires approximately O(N^2) operations. However, the proffered algorithm suggested by this study requires approximately O(N^3) operations. Hence, it can be stated that the existing NTRU has a better time complexity when compared with the proposed NTRU algorithm which is mainly as a result of the lattice multiplication operations.

5. Conclusion

This paper proposes a variant of NTRU cryptosystem with the focus to establish its fastness and security in a cloud environment. The proposed variant was simulated together with NTRU, RSA, AES and ECC cryptosystems to establish the time complexity of the algorithms in regard to key generation, encryption and decryption. The simulation showed that as in terms of the private and public key generation, the RSA cryptosystem showed to consume the least time (average). Similarly, in terms of key generation, comparing the existing and proposed NTRU cryptosystem, the existing NTRU cryptosystem proved to be more efficient for private and public key generation. More so, as regards encryption and decryption, the existing NTRU cryptosystem proved to be more efficient. The proposed NTRU cryptosystem has a lower throughput when compared with the existing NTRU algorithm which proposes that the existing NTRU has lower power consumption.

The introduction of lattice arithmetic to drive the processing of the existing NTRU cryptosystem via simulation has proved not to be effective.

6. References

[1] M. Marwan, A. Kartit, and H. Ouahmane, (2017), “A Secured Data Processing Technique for Effective Utilization of Cloud Computing”, Journal of Data Mining and Digital Humanities. Special Issue on Scientific and Technological Strategic Intelligence.

[2] G. Summers, (2004). Data and databases. In: Koehne, H Developing Databases with Access: Nelson Australia Pty Limited. p4-5.

[3] A. J. Gabriel, B. K. Alese, A. O. Adetunmbi, O. S. Adewale (2015), “Post-quantum crystography based security framework for Cloud Computing”, Journal of Internet Technology and Secured Transactions Vol 4 Issue 1 351-357.

[4] T. Sanamrad, (2014), “Encrypting Databases in the Cloud, Threats and Solutions ETH Zurich, Switzerland”. http://e-collection.library.ethz (Access DateL 15 November, 2021).

[5] S. Xue, and C. Ren, (2019), “Security Protection of System Sharing Data with Improved CP-ABE Encryption Algorithm under Cloud Computing Environment”, Autom. Control Comput. Sci. 53, 342–350.

[6] H. Huang, Y. Zhao, T. Li, F. Li, Y. Du, F. Xiang-Qun S. Zhang, X. Wang, and B. Wan-Su, (2016), “Performing Homomorphic Encryption Experiments on IBM’s Cloud Quantum Computing Platform”, Available at: https://arxiv.org › cs. (Access Date: 2 April 2021).

[7] D. J. Bernstein, J., Buchmann, and E. Dahmen, (2009). Introduction to post-quantum cryptography. (Introductory chapter to book "Post-quantum cryptography”). Springer, Germany.

[8] D. J. Bernstein, N. Heninger, P. Lou, and L. Valenta, (2017), “Post-quantum RSA”, Available at: https://cr.yp.to/papers/pqrsa-20170419.pdf. (Access Date: 4 June 2018).

[9] A. Thompson, O.E. Oyinloye, M.T. David, and B.K. Alese, (2020), “A Secured System for Internet Enabled Host Systems. Network and Commination Technologies”, Vol. 5, No 1. DOI: 10.5539/nct.v5n1p26.

[10] A. M. Kuo, (2011), “Opportunities and Challenges of cloud computing to improve health care services”, https://www.ncbi.nlm.nih.gov/pmc/articles/pmc3222190/. (Access Date: 2 March 2018).

[11] F. Thabit, S. Alhomdy, S., Abdulrazzaq, H. A. Ahdal, and S. Jagtap, (2021), “A new lightweight cryptographic algorithm for enhancing data security in cloud computing”, Global Transitions Proceedings.

[12] S. Chandel, G. Yang, and S. Chakravarty, (2020), “RSA-CP-IDABE: A Secure Framework for Multi-User and Multi-Owner Cloud Environment”, Information. 11, pp 382.

[13] I. J. Awan, M. Shiraz, M. U. Hashmi, Q. Shaheen, R. Akhtar, and A. Ditta, (2020), “Secure Framework Enhancing AES Algorithm in Cloud Computing”, Hindawi, Security and Communication Networks. Volume 2020, https://doi.org/10.1155/2020/8863345.

[14] R. Kumar, A. S. Naidu, A. Singh, and A. N. Tentu (2020), “McEliece cryptosystem: simulation and security vulnerabilities”, Int. J. Computing Science and Mathematics, Vol. 12, No. 1 pp 64–81.

[15] N. Rani, N. Juliet and S. Arunkumar (2020), “A Novel Cryptosystem for Files Stored in Cloud using NTRU Encryption Algorithm” International Journal of Recent Technology and Engineering (IJRTE). 2277-3878 Volume-9 Issue.

[16] K. Ramkumar, and G. Gunasekaran, (2019), “Preserving security using crisscross AES and FCFS scheduling in cloud computing”, Int. J. Advanced Intelligence Paradigms, Vol. 12, Nos. 1/2.

[17] M. Kindberg, (2017), “A usability study of post-quantum algorithms”, A Masters thesis submitted to the Department of Electrical and Information Technology, Lund University.

[18] Z. Balogh, and M. Turcani, (2016), “Modeling of Data Security in Cloud Computing”, Available at: https://www.researchgate.net/publication/319509441_An_Overview_on_Data_Security_in_Cloud_Computing. (Access Date: 26 Febraury 2018).

[19] M. Abdelnapi, F. A. Omara, and N. F. Omran (2016), “A Hybrid Hashing Security Algorithm for Data Storage on Cloud Computing”, International Journal of Computer Science and Information Security (IJCSIS), Vol. 14, No. 4.

[20] S. Pavithra, and S. Baskar, (2015), “Lattice based Multiplier for WSN Applications for ECC”, International Journal of Trend in Research and Development, Vol. 2(6).

[21] A. J. Gabriel, B. K. Alese, A. O. Adetunmbi, O. S. Adewale, (2013), “Post-Quantum Crystography: A Combination of Post-

Quantum Cryptography and Steganography". The 8th International Conference for Internet Technology and Secured Transactions (ICITST-2013), Technically Co-sponsored by IEEE UK/RI Computer Chapter, 9th-12th December 2013, London, UK, pp 454-457.

[22] A. J. Gabriel, A. Darwish, A. E. Hassanien (2021), "Cyber Security in the Age of COVID-19", In: Hassanien A.E., Darwish A. (eds) Digital Transformation and Emerging Technologies for Fighting COVID-19 Pandemic: Innovative Approaches. Studies in Systems, Decision and Control, vol 322. Springer, Cham.

[23] H. C. Ukwuoma, A. J. Gabriel, A. F. Thompson and B. K. Alese, (2021), "Optimised Privacy Model for Cloud Data," 2021 16th International Conference on Computer Science and Education (ICCSE), 2021, pp. 267-269, DOI: 10.1109/ICCSE51940.2021.9569395.

[24] A. J. Gabriel, B. K. Alese, A. O. Adetunmbi, O. S. Adewale, O. A. Sarumi (2019). "PostQuantum Crystography System for Secure Electronic Voting". Open Computer Science, DeGruyter; 9:292-298. DOI: 10.1515/comp-2019-0018.

[25] N. Gama, N. Howgrave-Graham, and P. Q. Nguyen (2006), "Symplectic Lattice Reduction and NTRU. In: Vaudenay S. (eds) Advances in Cryptology" - EUROCRYPT 2006. Lecture Notes in Computer Science, vol 4004. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11761679_1.

[26] A. Zalekian, M. Esmaeildoust, and A. Kaabi, (2015), "Efficient Implementation of NTRU Cryptography using Residue Number System". International Journal of Computer Applications (0975 – 8887) Volume 124 – No.7.

[27] N. Mishra, T. K. Sharma, V. Sharma and V. Vimal, (2018), "Secure Framework for Data Security in Cloud Computing", Soft Computing: Theories and Applications, Advances in Intelligent Systems and Computing 583, https://doi.org/10.1007/978-981-10-5687-1_6.

[28] R. Lizy, and V. Raj (2021), "Improvement of RSA Algorithm Using Euclidean Technique", Turkish Journal of Computer and Mathematics Education. Vol.12 No.3.

[29] D. Elminaam, H. Kader, and M. Hadhoud, (2009), "Performance Evaluation of Symmetric Encryption Algorithms", IJCSNS International Journal of Computer Science and Network Security. Vol.8 No.12.

[30] E. Camacho-Ruiz, M. C. Martínez-Rodríguez, S. Sánchez Solano and P. Brox, "Accelerating the Development of NTRU Algorithm on Embedded Systems," 2020 XXXV Conference on Design of Circuits and Integrated Systems (DCIS), 2020, pp. 1-6, doi: 10.1109/DCIS51330.2020.9268647.

[31] E. Malekian, A. Zakerolhsooeini, (2010), "OTRU: A non-associative and high-speed public key cryptosystem", IEEE Computer Society, 83–90.

[32] H.H. Abo-Alsood and H.R. Yassein, (2021), "Design of an alternative NTRU Encryption with High Secure and Efficient", International Journal of Mathematics and Computer Science, No 4 1469-1477.

[33] R. Adee, R. and H.A. Mouratidis, (2022), "Dynamic Four-Step Data Security Model for Data in Cloud Computing Based on Cryptography and Steganography", Sensors 2022, 22, 1109. https://doi.org/10.3390/s22031109.

[34] S. Kumar, G. Kamani, M.S. Guar, A. Mishra, (2021), "Cloud Security using Hybrid Cryptography Algorithms", 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM).

[35] R. Sugumar and K. Raja (2018), "EDSMCCE: Enhanced Data Security Methodology for Cloud Computing Environment", International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Vol. 3, Iss 3.

[36] K.S. Suresh (2021), "What are Quantum Cryptography's disadvantages", Available at https://www.what-are-quantum-cryptography's-disadvnatages. Access Date: 16th February, 2022.

[37] F. Wang, H Wang, L. Xue (2021), "Research on Data Security in Big Data Cloud Computing Environment", 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC) | 978-1-7281-8028-1/20/$31.00 ©2021 IEEE | DOI: 10.1109/IAEAC50856.2021.9391025.