Comparative Analysis of RSA and NTRU Algorithms and Implementation in the Cloud

Bambang Harjito, Henny Nurcahyaning Tyas, Esti Suryani, Dewi Wisnu Wardani
Department of Informatics, Universitas Sebelas Maret
Jl. Ir. Sutami 36A Kentingan, Surakarta, Indonesia

***Abstract*—The emergence of cloud computing platforms makes it easier to connect and collaborate globally without setting up additional infrastructures such as servers and data centers. This causes the emergence of threats to data security against digital information. This security threat can be overcome by cryptography. Examples of cryptographic algorithms are RSA and NTRU. The main concern that arises in this research is how to perform a comparative analysis between asymmetric cryptographic algorithms, RSA (Rivest-Shamir-Adleman) and NTRU (Nth-Degree Truncated Polynomial Ring) algorithms and their implementation in cloud storage. Comparison of performance between the RSA and NTRU algorithms at security levels 80, 112, 128, 160, 192, and 256 bits by running 5 – 1000 data, the results obtained that the running time of the key generation process and encryption of the NTRU algorithm is more efficient than the RSA algorithm. Wiener's Attack test on the RSA algorithm and LLL Lattice Basis Reduction on the NTRU algorithm. NTRU algorithm has a more secure level of resilience, so that it can be said that the NTRU algorithm is more recommended for cloud storage security.**

Keywords*—Attacks; privacy; cryptography; RSA; NTRU; cloud storage

I. Introduction

Analysis of more than 135,000 organizations in 2020 shows that globally, cloud adoption has reached 81% as measured by the use of productivity platforms based on research from an information technology company and US research firm Gartner. Cloud services have become a significant industry. Cloud expects to grow from USD 50.1 billion in 2020 to USD 137.3 billion by 2025, with a Compound Annual Growth Rate (CAGR) of 22.3% research conducted Markets and Markets analysis. The increasing popularity of the cloud is also accompanied by several security problems in the cloud that are vulnerable to the possibility of being exposed to unwanted parties, especially security breaches in cloud storage [1]. Cryptography is one of the most effective and efficient components of network security in securing information. Cryptography ensures that only the party who has exchanged the keys can read the encrypted message (authentic party) [2]. The RSA (Rivest-Shamir-Adleman) and NTRU (Nth-Degree Truncated Polynomial Ring) algorithms are asymmetric cryptographic systems, however the NTRU algorithm is a lattice-based algorithm where the key selection is not only strong but also difficult to solve. Grid-based cryptography in general, is an improvement over classical number theory algorithms such as the RSA algorithm [3], [4]. This system is also known for its high level of security based on worst-case hardness. Worst-case hardness is based on the complexity of the problem grids and the shortest vector problem (SVP). NTRU cryptosystem is a lattice-based cryptography known to withstand quantum computing attacks, and classical computing attacks [5], [6]. The RSA algorithm is more optimal for the encode process than the DES algorithm [7] besides that the RSA algorithm is superior to the ElGamal algorithm [8].

Each algorithm bases on a different problem, such as the security of the RSA algorithm, which is difficult to factor large numbers into prime factors [9]. The operation of the NTRU algorithm is based on objects in a truncated polynomial ring R= Z[X]/(X^N-1) with convolution multiplication [10], [11]. The RSA algorithm, which is based on integer factoring, can be attacked using the Wiener's Attack algorithm, while the NTRU algorithm can be attacked with the well-known algorithm to find short vector, LLL (Lenstra-Lenstra-Lovasz) Lattice Basis Reduction algorithm, which is a lattice-based reduction algorithm [12]. In this research, a comparative analysis will be carried out between the RSA (Rivest-Shamir-Adleman) and NTRU (Nth-Degree Truncated Polynomial Ring) cryptographic algorithms and obtain the running time of key generation, encryption, decryption, and security level and see which algorithm is better and implementation on cloud storage using Flask.

II. Related Work

The RSA algorithm is an algorithm that can be used to maintain the security and confidentiality of fingerprint data. Besides that, the RSA algorithm can be also applied to cloud computing security by using digital signatures combined with the AES algorithm [13], [14]. The NTRU algorithm is a lattice-based algorithm which refers to the lattice-based algorithm that makes the NTRU algorithm more resistant to quantum computing attacks [5], [6], [15], [16]. Cloud refers to a set of services and infrastructure accessed through the internet. Cloud service providers must use encryption algorithms to protect user data, such as the use of Advanced Encryption Standard (AES) algorithms, Rivest–Shamir–Adleman (RSA), Elliptic Curve Cryptography(ECC), and NTRU Encryption [17]–[19], in addition to the cloud, the application of the RSA and NTRU algorithms can implement in the IoT environment, the use of an accelerated IoT environment, the security of data collected and stored by devices becomes important [20]. NTRU's lattice-based cryptosystem has lattice problems in the form of grid problems and Shortest Vector Problem (SVP) and can be attacked using the LLL (Lenstra-Lenstra-Lovasz) algorithm, a well-known lattice-based reduction algorithm for grid-based

cryptographic grid problems [21]–[24]. The RSA cryptosystem is the most used in the SSL/TLS protocol that allows sensitive information to be sent via the internet. Wiener's Attack shows that the RSA algorithm can be attacked when the value of d is relatively small compared to the modulus of N [25].

III. Proposed Work

In this section, we provide an overview of our solution comparative analysis of RSA and NTRU Algorithm and Implementation in the Cloud. The comparative analysis model is depicted in Fig. 1. According to the model, the model consist of five steps : (*A*) the key generation, encryption decryption process of RSA, (*B*) the key generation, encryption, decryption process of NTRU (*C*) Winners attacks on the RSA (*D*) LLL attack on and NTRU, (*E*) Implemented in cloud storage. The process begins with the user who will upload the file to cloud storage, this file can be called plaintext. The encryption process will use the RSA and NTRU algorithm public keys. After the encryption process is finished, an encrypted message or ciphertext will be obtained, which will then be uploaded to cloud storage, then calculated and compared for the running time of the public key generation, the private key, and the running time of the encryption process, and the attack process on the public key. After the ciphertext file is uploaded to cloud storage, a download process will be carried out, where this process will decrypt the encrypted message using a private key and will obtain a decrypted file or plaintext file, then compare the running time of the decryption process on the RSA and NTRU algorithms.

A. RSA (Rivest-Shamir-Adleman)

In this section, the key generation, encryption decryption process of RSA is the first step. It can be explained as follows:

1) Key generation: The key generation process in the RSA algorithm begins with selecting the prime numbers p and q, then looking for the value of $n = p * q$. Then select the e, $e < \phi$, where $\phi(n) = (p-1) * (q-1)$, must be a prime number. Select the encryption key 'e', $1 < e < \phi(n)$,

$$gcd\,(e, \phi(n)) = 1 \quad (1)$$

e and $\phi(n)$ are coprime. By using the Expanded-Euclidean algorithm to calculate d, then:

$$ed \equiv 1(mod\ \phi(n)) \quad (2)$$

Or

$$ed \equiv k\phi(n) + 1 \quad (3)$$

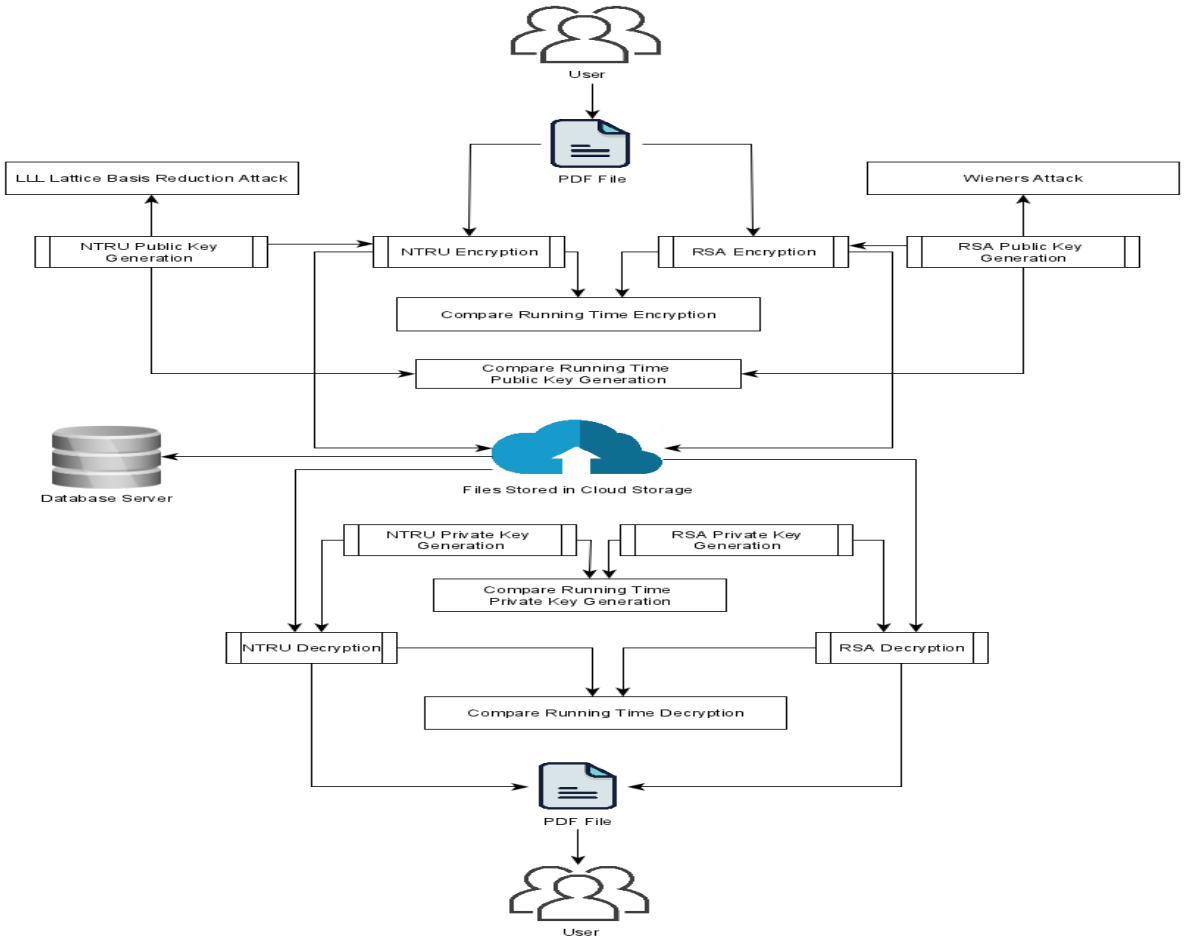

Fig. 1. Comparative Analysis of RSA and NTRU Algorithm and Implementation in the Cloud.

Testing a number with the Rabin-Miller algorithm. If b_0 = 1 or $b_0 = n - 1$, then n is a prime number.

$$b_0 = a^m (mod\ n) \quad (4)$$

If $b_0 \neq 1$ or $b_0 \neq n - 1$, do with Equation (9).

$$b_1 = {b_0}^2 mod\ n \quad (5)$$

If $b_1 = n - 1$, then b_1 validated probability as a prime.

2) RSA encryption: Plaintext is made into blocks of $m_1, m_2, m_3, \ldots, m_n$ so that each block represents a value between $[0, n-1]$ or $(0 < m_1 < n-1)$. Where input message $m < n$. Then calculate the c_i ciphertext block for the plaintext block through the Equation (6).

$$c_i = {m_i}^e\ mod\ n \quad (6)$$

3) RSA decryption: Ciphertext c_i is processed using Equation (7) to get the original message, the plaintext message.

$$m_i = {c_i}^d\ mod\ n \quad (7)$$

B. NTRU (Nth-Degree Truncated Polynomial Ring)

The key generation, encryption decryption process of NTRU is the second steps. Before explaining further from the stage one to the next. We give the principle of NTRU.

The principle of the object used by the NTRU public-key cryptosystem is to use a polynomial of degree $N - 1$. If a and b are two polynomials in the ring R, they can be defined in Equation (8) and (11).

$$a = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_{N-2} x^{N-2} + a_{N-1} x^{N-1} \quad (8)$$

$$= \sum_{i=0}^{N-1} a^i x^i \quad (9)$$

Coefficient vector$a$ will be represented as in the Equation (10).

$$a = (a_0, a_1, a_2, \ldots, a_{n-2}, a_{N-1}) \quad (10)$$

$$b = b_0 + b_1 x + b_2 x^2 + b_3 x^3 + \cdots + b_{N-2} x^{N-2} + b_{N-1} x^{N-1} \quad (11)$$

$$= \sum_{i=0}^{N-1} b^i x^i \quad (12)$$

Coefficient vector b will be represented as the Equation (13).

$$b = (b_0, b_1, b_2, \ldots, b_{n-2}, b_{N-1}) \quad (13)$$

The basic operations used in convoluted ring polynomials are addition, subtraction, and convolution multiplication. The polynomial coefficient (a_0, a_1, …, a_{n-1}) is an integer. Some coefficient values are 0. This set of polynomials is called R.

1) Key generation: NTRU Key Generation begins with selecting two polynomials $f \in \mathcal{L}_f$ and $g \in \mathcal{L}_g$ provided that f has an inverse modulo p and q, so f_p and f_q can writing in Equation (14), (15).

$$f * f_p \equiv 1\ (mod\ p) \quad (14)$$

$$f * f_q \equiv 1\ (mod\ q) \quad (15)$$

The private key consists of the polynomials f and f_p. After determining the polynomials f and g, the public key can be calculated by the Equation (16).

$$h \equiv p f_q * g\ (mod\ q) \quad (16)$$

2) NTRU encryption: To perform the encryption process, one chooses a polynomial m representing a message so that $m \in Lm$ and a random polynomial $r \in Lr$. The message must be converted to a polynomial m. Then select a small random polynomial, $r \in R$ used to shuffle the messages. Calculating the ciphertext e, with the Equation (17).

$$e \equiv r * h + m\ (mod\ q) \quad (17)$$

3) NTRU decryption: The decryption process begins by calculating the polynomial $a = f * e\ (modulo\ q)$, then define the coefficient a between$-q/2$ and $q/2$, then calculating the polynomial $b = a\ (modulo\ p)$ so that the private key f_p is obtained to calculate the value of d.

$$d = f_p * b\ (mod\ p) \quad (18)$$

Or

$$d \equiv f_p * [f * e]_q (mod\ p) \quad (19)$$

C. Wiener's Attack

In this section, Wiener's attack is a type of cryptographic attack against the RSA algorithm. This attack is the third steps. The attack uses an advanced fraction method (continued fraction). Continued Fraction of rational number $\frac{u}{v}$ is an expression of the form $x = a_0$, and we get Equation (20).

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \ldots}}} \quad (20)$$

Where the coefficient a_0 is an integer and all other coefficients for a_i for $i \geq 1$ are positive integers. The coefficient a_i is called the partial quotients of the continued fraction. Generate unique continued fraction, with Euclidean algorithm, can efficiently determine all coefficients $a_0, a_1, \ldots, a_N$.

D. LLL Lattice basis Reduction

LLL attack on and NTRU is the four steps. It can be depicted in Fig. 2.

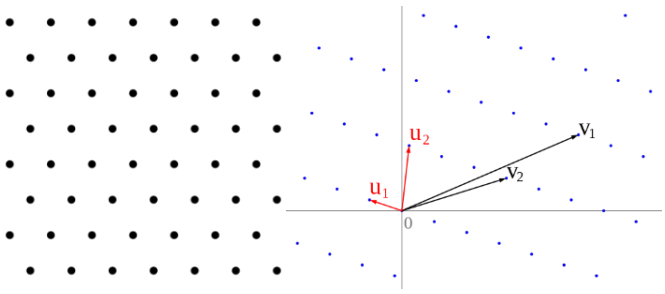

Fig. 2. Basis Lattice.

Fig. 2 shows an image of the vector lattice and lattice, which is the basis of the LLL algorithm. The LLL algorithm is a lattice basis reduction algorithm. The basis of a lattice$B = \{b_1, b_2, \ldots, b_n\}$ will be defined as Gram-Schmidt basis $B = \{b_1^*, b_2^*, \ldots, b_n^*\}$ by fulfilling the two conditions below:

(Size reduction) :

$$| \mu_i j | = \frac{\left|b_i . b_j^*\right|}{\left\|b_j^*\right\|^2} \leq \frac{1}{2}, where\ 1 \leq j < i \leq n. \quad (21)$$

(Lovasz Condition) :

$$\|b_i^*\|^2 \geq \left(c - \mu_{i,i-1}^2\right)\|b_{i-1}^*\|^2, for\ \frac{1}{4} < c < 1\ dan\ 1 < i < n. \quad (22)$$

E. Cloud Storage

In this section, implemented in cloud storage is the five steps. Cloud storage is a cloud computing model that stores data on the internet through a cloud computing provider. *Cloud , storage* is a cloud computing system that allows users to store and share data on the internet [26]. Cloud storage operates online, making it easier to retrieve and manage data. A cloud storage architecture, where a web browser will connect to a server that automatically accesses the database server. It can be depicted in Fig. 3.

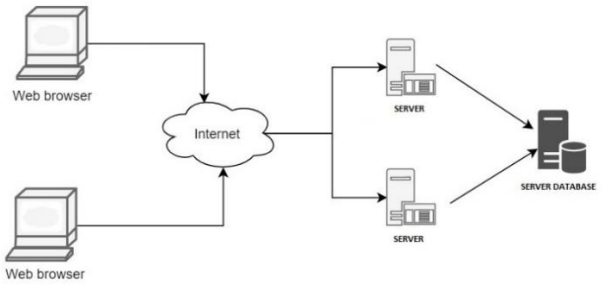

Fig. 3. Cloud Storage Architecture.

IV. Performance Analysis

In this section, we perform a comparative analysis of RSA and NTRU Algorithm and an Implementation in the Cloud.

A. Parameter

Security level algorithms RSA and NTRU are used in the Key Generation process. Table I shows the corresponding NTRU and RSA key sizes with equivalent security levels. Security level (k) 80, 112, 128, 160, 192, and 256 bits [27], [28].

The parameter used in the RSA algorithm is to take advantage of the number of bits selected, where the value of these bits will affect the length of the key. The parameter selection of the NTRU algorithm used in this study is the value of N, p, and q. This value will be used as a determinant of the length of the public key and private key which will later be used for the encryption and decryption process. The magnitude of this parameter is obtained from research [27], [28]. The explanation of the parameter values for each security level is presented in Table II. As well as the results of the running time obtained based on the parameters used. It can be shows in Table III.

Table I and Fig. 4 show that NTRU's bandwidth usage is more efficient than RSA's when the level of security increases, from the same standard used in both RSA and NTRU algorithms, this security level will be used to compare the two algorithms so that we get a better result.

B. RSA (Rivest-Shamir-Adleman)

1) Key generation: From the process of key generation, encryption, and decryption of the RSA algorithm, the results are shown in Table II, showing the generation of public and private keys. The running time results are almost the same for low and moderate security, but public key generation is faster at the highest security. Private key generation on the RSA algorithm is faster at standard and high security.

2) RSA encryption and decryption: The RSA algorithm encryption process uses the solution $c_i = {m_i}^e \ mod\ n$, to obtain an encrypted message in the form of a ciphertext message. The process of decrypting the ciphertext message is done using the solution $m_i = {c_i}^d \ mod\ n$ to make the original message a plaintext message. The running time for the encryption and decryption of the RSA algorithm at each security level can be seen in Table II, showing that the results of the decryption process's running time are faster for each security level.

TABLE I. Security Level RSA, NTRU

Security Level (bits)	NTRU (bits)	RSA (bits)
80	2008	1024
112	3003	2048
128	3501	3072
160	4383	4096
192	5193	7680
256	7690	15360

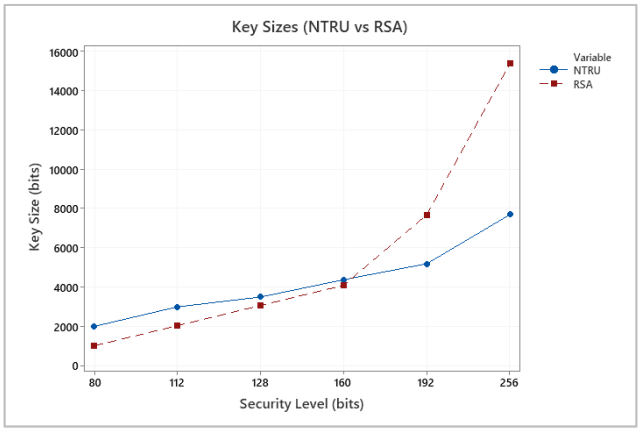

Fig. 4. Comparison of Security Level RSA, NTRU.

TABLE II. RUNNING TIME RSA

Security level	Security level (bits)	RSA (bits)	Public Key Generation (second)	Private Key Generation (second)	Encryption (second)	Decryption (second)
Low security	80	1024	0.62634	0.62634	0.0469	0.0153
Moderate security	112	2048	6.59910	6.59910	0.2594	0.0780
Standard security	128	3072	25.77820	25.77820	0.4458	0.1370
Standard security	160	4096	58.39774	58.39774	0.9420	0.2884
High security	192	7680	682.09213	682.09213	5.6284	1.7493
Highest security	256	15360	6920.16406	6920.16406	41.5171	12.5220

C. NTRU (Nth-Degree Truncated Polynomial Ring)

1) Key generation: The main parameters of the NTRU algorithm are integers N, p, q. This parameter value is used to determine polynomial rings. The results of the running time of the public key generation and the private key of the NTRU algorithm can be seen in Table III. Table III shows that the speed of the public key running time is faster than the private key at each security level.

2) NTRU encryption and decryption: The NTRU algorithm encryption process is carried out using the$e \equiv r * h + m_i \ (mod\ q)$ solution to obtain the ciphertext message. Ciphertext message can be processed with decryption using the NTRU algorithm to get plaintext message, with the solution $d \equiv f_p * [f * e]_q (mod\ p)$. The encryption and decryption process in the NTRU algorithm obtained results, as shown in Table III. It can be seen that the encryption and decryption process in the NTRU algorithm does not show a significant difference. However, it can be seen that the encryption process is faster for the low-security level and the highest security. Moderate, standard, and highest security indicate that the decryption process is faster than the encryption process. The security level and the value of the NTRU parameter increase affect the running time speed.

D. Comparison of RSA and NTRU Algorithms

1) Key generation: Longer keys will provide higher security but will consume more computing time, so the value of security and speed will be inversely related. Generating a key with a long bit size can take from a few minutes to several hours, as shown in Tables II and III. From the two tables, Table II and Table III, the results are that private and public key generation in the NTRU algorithm is much faster than key generation in the algorithm RSA for security levels 80, 112, 128, 160, 192, 256 bits, as depicted in Fig. 5.

2) Encryption: The time required to encrypt files using both algorithms is compared to evaluate system performance. From the data obtained, the time for encryption using RSA is faster than NTRU with an average speed of RSA encryption of 2.3285 seconds and described in detail at different security levels as shown in Table II and Table III.

TABLE III. RUNNING TIME NTRU

Security level	Security level (bits)	Key Sizes (bits)	N	p	q	Public Key (second)	Private Key (second)	Encryption (second)	Decryption (second)
Low security	80	2008	251	3	2048	0.5719	0.6281	0.5478	0.74925
Moderate security	112	3033	401	3	2048	1.9941	2.1696	1.5158	1.4010
Standard security	128	3501	439	3	2048	2.4076	2.6754	1.7236	1.6765
Standard security	160	4383	487	3	2048	2.9953	3.4613	3.6976	3.4571
High security	192	5193	593	3	2048	4.4289	5.1424	6.3630	5.4323
Highest security	256	7690	743	3	2048	7.9426	9.0723	7.9774	9.1061

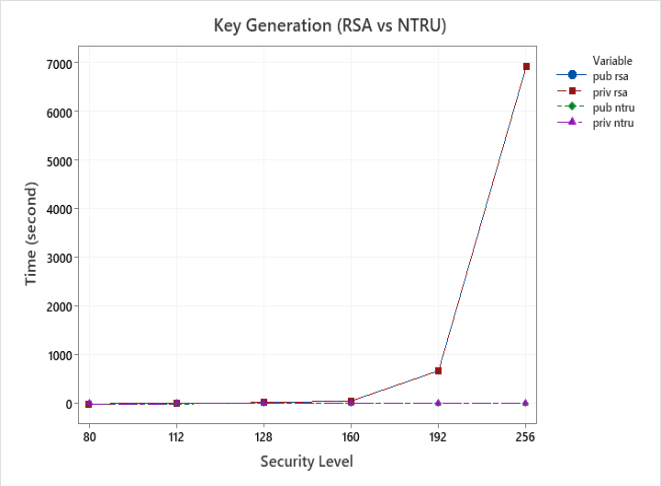

Fig. 5. Comparison of RSA, NTRU Key Generation.

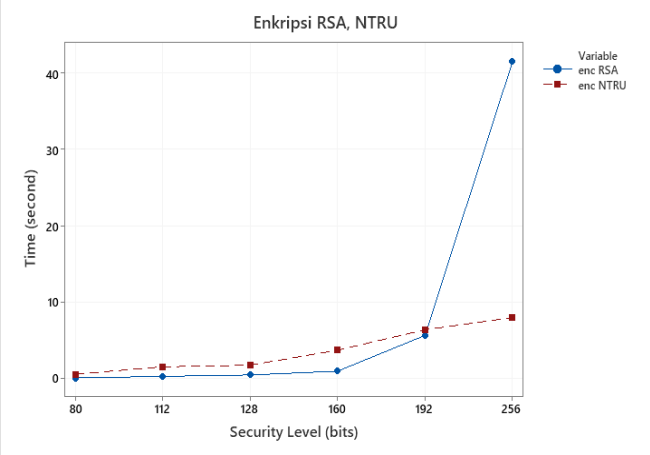

Fig. 6. Comparison of RSA, NTRU Encryption.

From Table II and Table III shows graphs of running time for the encryption process for the RSA and NTRU algorithms, as shown in Fig. 6. It can be seen that the encryption process for low-level RSA security is faster than NTRU, but when it reaches the security level of 256 bits, the NTRU algorithm is much faster than RSA.

3) Decryption: The average speed of the RSA algorithm decryption process is faster than the NTRU algorithm. RSA algorithm is more efficient in decrypting data than the NTRU algorithm. Table II and Table III show that when the security level is 80, 112, 128, 160, 192 bits, the RSA decryption process is faster, but when the security level is 256 bits, the NTRU process is faster than RSA decryption.

A comparison for the decryption process in Table II and Table III, a comparison chart is obtained for each security level as shown in Fig. 7. It can be seen that the RSA decryption process for low-security levels is faster than NTRU decryption, but when the security level is at the highest security level, which is 256 bits NTRU algorithm has a faster speed.

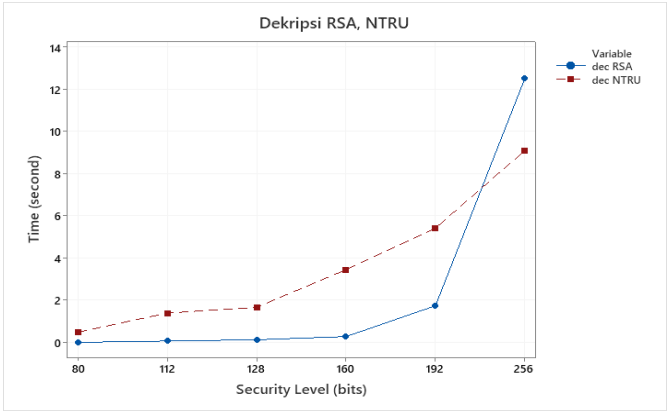

Fig. 7. Comparison of RSA, NTRU Decryption.

4) Wiener's attack on RSA: The security of the RSA algorithm depends on the difficulty of factoring large integers to obtain prime numbers, which is one of the mathematical computational problems that are difficult to solve. [29]. Wiener's Attack algorithm uses a continued fraction solution. Table IV is the result of the running time of the Wiener's attack process at different security levels, which shows that when the security level is increased, the time to carry out attacks will increase.

5) LLL lattice basis reduction on NTRU: The NTRU algorithm with parameters N, p, q can show each parameter's security level. The public key on the NTRU algorithm will be tested using an attack in the form of LLL (Lenstra-Lenstra-Lovasz) lattice basis reduction with the output obtained in the form of running time. This attack can show the strength of the NTRU algorithm. The strength of the NTRU algorithm is in the difficulty of finding a short vector of a lattice.

In Table V, it can be seen how the LLL lattice base reduction algorithm runs on different parameters. When the value of parameter N increases, the time to carry out an attack also increases, and when the value of N = 31, the time required to carry out an attack takes more than 9 hours. From Table V, it can be said that LLL can perform attacks on the NTRU algorithm but in small parameters and cannot find short vector problems (SVP) for a larger basis. When N's value is higher, the running time of the attack process using LLL tends to increase.

TABLE IV. RUNNING TIME WIENER'S ATTACK

Security level	Security level (bits)	RSA (bits)	Wieners Attack (second)
Low security	80	1024	0.09733
Moderate security	112	2048	0.35923
Standard security	128	3072	1.19537
Standard security	160	4096	1.46252
High security	192	7680	5.87838

TABLE V. RUNNING TIME LLL ATTACK

No	N	p	q	LLL Attack (second)
1	24	3	128	6780
2	25	3	128	7260
3	26	3	128	18600
4	27	3	128	15420
5	28	3	128	24060
6	29	3	128	28980
7	31	3	128	> 9 hours (32400 second)

E. Implementation on Cloud Storage

1) Upload process: Fig. 8 is a flow block diagram of the file upload process and when it is implemented to cloud storage, as shown in Fig. 9. Fig. 9 shows the results of file uploads using the RSA and NTRU algorithms applied to file storage. When the uploaded file has been selected, the uploaded file will be converted into a ciphertext message then sent to the database server.

2) Download process: Fig. 10 is a block diagram of the file download process that can be implemented in cloud storage. Fig. 9 shows the uploaded file can be downloaded and the file decryption process by pressing the download button.

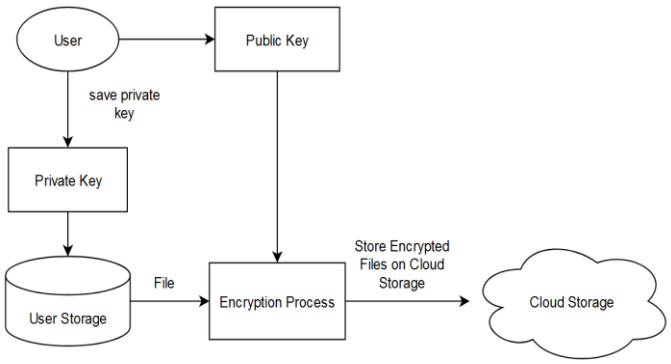

Fig. 8. Block Diagram Upload File.

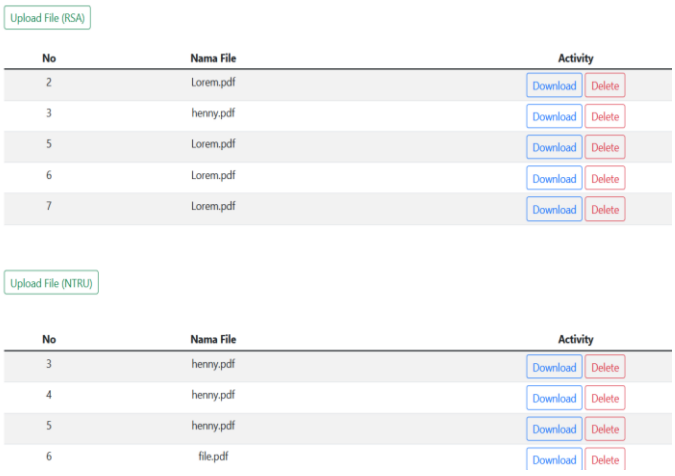

Fig. 9. File Storage.

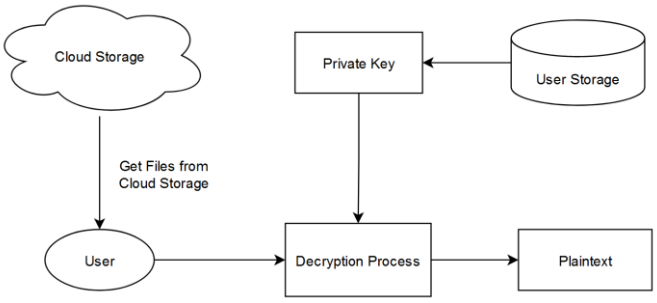

Fig. 10. Block Diagram Download File.

After pressing the download button, the program will download the file and do the decryption process. Then will get the result of a plaintext message.

V. CONCLUSION

The RSA algorithm (Rivest-Shamir-Adleman) and the NTRU algorithm (Nth-Dimensional Truncated Polynomial Ring) are algorithms to secure plaintext or original messages by encrypting messages. In this study, the two algorithms compared their performance in key generation, encryption, decryption, attack, and their implementation in cloud storage. Performance comparisons are made with two things, measuring running time and testing the security of attack attempts on both algorithms. From the results of this study, the results are as in Table II and Table III. The use of the selected parameter for the RSA bit, the higher the bit selected, the greater the time required. The greater the value of the N parameter in the NTRU algorithm, the greater the time required for the key generation, encryption, and decryption processes.

In terms of running time in the key generation and encryption process, the NTRU algorithm is more efficient than the RSA algorithm. In terms of security, by testing the Wiener's attack on the RSA algorithm and LLL Lattice Basis Reduction on the NTRU algorithm, it shows that the NTRU algorithm has a more secure level of resilience so that it can be said that the NTRU algorithm is more recommended for cloud storage security. In this paper, we have not discussed the comparison of the LLL algorithm attacks applied to the RSA algorithm and the NTRU algorithm. The comparison analysis of the RSA algorithm and the NTRU algorithm has proven successful, but it is hoped that in future research a different and updated implementation can be carried out using other algorithms, such as the comparison of the ECC and Elgamal algorithms.

REFERENCES

[1] M. Ahmed and M. Hossain, "Cloud Computing and Security Issues in the Cloud," International Journal of Network Security & Its Applications, vol. 6, pp. 25–36, Jan. 2014, doi: 10.5121/ijnsa.2014.6103.

[2] V. Mavroeidis, K. Vishi, M. D. Zych, and A. Jøsang, "The Impact of Quantum Computing on Present Cryptography," ijacsa, vol. 9, no. 3, 2018, doi: 10.14569/IJACSA.2018.090354.

[3] E. Crockett, "Simply safe lattice cryptography," Jul. 2017, Accessed: Oct. 17, 2021. [Online]. Available: https://smartech.gatech.edu/handle/1853/58734.

[4] L. Ducas, "Advances on quantum cryptanalysis of ideal lattices," undefined, 2017, Accessed: Oct. 17, 2021. [Online]. Available: https://www.semanticscholar.org/paper/Advances-on-quantum-

cryptanalysis-of-ideal-lattices-Ducas/bca6ef077421a07276f1b98623fe663e89e515da.

[5] Z. Jing, C. Gu, Z. Yu, P. Shi, and C. Gao, "Cryptanalysis of lattice-based key exchange on small integer solution problem and its improvement," Cluster Comput, vol. 22, no. 1, pp. 1717–1727, Jan. 2019, doi: 10.1007/s10586-018-2293-x.

[6] Z. Liu, K.-K. R. Choo, and J. Grossschadl, "Securing Edge Devices in the Post-Quantum Internet of Things Using Lattice-Based Cryptography," IEEE Communications Magazine, vol. 56, pp. 158–162, Feb. 2018, doi: 10.1109/MCOM.2018.1700330.

[7] M. N. A. Wahid, B. Esparham, A. Ali, and M. Marwan, "A Comparison of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish for Guessing Attacks Prevention," 2018, [Online]. Available: https://symbiosisonlinepublishing.com/computer-science-technology/computerscience-information-technology32.pdf.

[8] Dr. M. Subhashini and Dr. R. Gopinath, "MAPREDUCE METHODOLOGY FOR ELLIPTICAL CURVE DISCRETE LOGARITHMIC PROBLEMS – SECURING TELECOM NETWORKS," 2020, doi: 10.34218/IJEET.11.9.2020.026.

[9] W. Susilo and J. Tonien, "A Wiener-type attack on an RSA-like cryptosystem constructed from cubic Pell equations," 2021, Accessed: Feb. 05, 2022. [Online]. Available: https://sci-hub.ee/https://doi.org/10.1016/j.tcs.2021.06.033.

[10] B. Santhiya and K. Anitha Kumari, "Analysis on DGHV and NTRU Fully Homomorphic Encryption Schemes," in Proceedings of International Conference on Artificial Intelligence, Smart Grid and Smart City Applications, Cham, 2020, pp. 669–678. doi: 10.1007/978-3-030-24051-6_61.

[11] X. Shen, Z. Du, and R. Chen, "Research on NTRU Algorithm for Mobile Java Security," Jan. 2009, pp. 366–369. doi: 10.1109/EmbeddedCom-ScalCom.2009.72.

[12] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients," Math. Ann., vol. 261, no. 4, pp. 515–534, Dec. 1982, doi: 10.1007/BF01457454.

[13] A. B. R. P. A. Putra, "Perbandingan Algoritma Rsa dan Elgamal pada Keamanan Data Sidik Jari," 2017, Accessed: Jan. 12, 2021. [Online]. Available: https://digilib.uns.ac.id/dokumen/76770/Perbandingan-Algoritma-Rsa-dan-Elgamal-pada-Keamanan-Data-Sidik-Jari

[14] A. L. KHOIRULLOH, "KEAMANAN FILE DALAM CLOUD COMPUTING DENGAN DIGITAL SIGNATURE ALGORITMA RSA DAN ENKRIPSI FILE ALGORITMA AES," 2019.

[15] D. J. Bernstein, J. Buchmann, and E. Dahmen, Eds., Post-Quantum Cryptography. Berlin Heidelberg: Springer-Verlag, 2009. doi: 10.1007/978-3-540-88702-7.

[16] J. Hoffstein, J. Pipher, and J. H. Silverman, "NTRU: A ring-based public key cryptosystem," in Algorithmic Number Theory, Berlin, Heidelberg, 1998, pp. 267–288. doi: 10.1007/BFb0054868.

[17] A. K. Bajwa and M. L. Sahi, "NTRU based Security in Cloud Computing," 2018, doi: 10.18535/IJECS/V7I6.09.

[18] O. Pandithurai, S. Meena S., R. Shenbagalakshmi, and A. U. Sindujha, "A Noval Approach of Drops with NTRU in Cloud," in 2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), Chennai, India, Mar. 2019, pp. 261–265. doi: 10.1109/ICONSTEM.2019.8918897.

[19] N. Suba Rani, "A Novel Cryptosystem for Files Stored in Cloud using NTRU Encryption Algorithm," IJRTE, vol. 9, no. 1, pp. 2127–2130, May 2020, doi: 10.35940/ijrte.A2536.059120.

[20] A. Nandanavanam, I. Upasana, and N. Nandanavanam, "NTRU and RSA Cryptosystems for Data Security in IoT Environment," in 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), Bengaluru, India, Oct. 2020, pp. 371–376. doi: 10.1109/ICSTCEE49637.2020.9277159.

[21] X. Deng, "An Introduction to Lenstra-Lenstra-Lovasz Lattice Basis Reduction Algorithm," p. 11, 2016.

[22] C. Peikert, "Lattice Cryptography for the Internet," in Post-Quantum Cryptography, vol. 8772, M. Mosca, Ed. Cham: Springer International Publishing, 2014, pp. 197–219. doi: 10.1007/978-3-319-11659-4_12.

[23] D. Stehlé and R. Steinfeld, "Making NTRU as Secure as Worst-Case Problems over Ideal Lattices," in Advances in Cryptology – EUROCRYPT 2011, vol. 6632, K. G. Paterson, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 27–47. doi: 10.1007/978-3-642-20465-4_4.

[24] D. Chi, J. W. Choi, J. S. Kim, and T. Kim, "Lattice Based Cryptography for Beginners," IACR Cryptol. ePrint Arch., 2015.

[25] W. Susilo, J. Tonien, and G. Yang, "A generalised bound for the Wiener attack on RSA," Journal of Information Security and Applications, vol. 53, p. 102531, Aug. 2020, doi: 10.1016/j.jisa.2020.102531.

[26] P. Yang, N. Xiong, and J. Ren, "Data Security and Privacy Protection for Cloud Storage: A Survey," IEEE Access, vol. 8, pp. 131723–131740, 2020, doi: 10.1109/ACCESS.2020.3009876.

[27] N. Howgrave-Graham, J. H. Silverman, and W. Whyte, "Choosing Parameter Sets for NTRUEncrypt with NAEP and SVES-3," in Topics in Cryptology – CT-RSA 2005, vol. 3376, A. Menezes, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 118–135. doi: 10.1007/978-3-540-30574-3_10.

[28] S. Chandel, W. Cao, Z. Sun, J. Yang, B. Zhang, and T.-Y. Ni, "A Multi-dimensional Adversary Analysis of RSA and ECC in Blockchain Encryption," in Lecture Notes in Networks and Systems, 2020, pp. 988–1003. doi: 10.1007/978-3-030-12385-7_67.

[29] C. Easttom, Modern Cryptography: Applied Mathematics for Encryption and Information Security. Springer International Publishing, 2021. doi: 10.1007/978-3-030-63115-4.