SECURE DATA STORAGE IN CLOUD COMPUTING USING QAES AND NTRU ALGORITHMS

Geetha Rani E[1], Chetana D T[2]

[1,2] GITAM University.
[1] grani@gitam.in, [2]ctukkojigitam.in

Abstract. Information must be kept private and secure because it is shared when people communicate online, making data protection essential. Data sharing is a significant element of this. Information can now be viewed by unauthorised interceptors due to the volume of data being transmitted online. Cryptography has been a key component of security systems. A blank message is encrypted during this process to add protection. An exact copy ensures sufficient protection of data in both conventional and quantum computing is urgently needed given the rise of quantum computing, as encryption is currently the most popular method of cloud data protection. Symmetric key cryptosystems, in comparison to public key cryptosystems, are faster because they only need a single private key to encode and decrypt data at both ends. Even so, it can be challenging to maintain security in a hostile environment while carrying out compatible and effective key distribution and secure private data transmission across organisations. In this paper comprehensive analysis of this cryptosystem is presented and describes the component-by-component approach used in its implementation. The different attacks on the McEliece cryptosystem are covered separately. The experimental results obtained using Goppa codes are also reported in the research where the simulations are carried out at different extension degrees. Using the results of the simulations, we came to our findings about the outcomes and the numerous implementation issues. In this project, a model is proposed that applies the Cloud customer data security using NTRUs (nth degree truncated polynomial ring units) together with a McEliece variation cryptosystem to secure access control data. To encrypt and decode data, the modified NTRU cryptosystem is employed., which is powered by lattice math. The process of multiplying larger numbers or conducting complex multiplication is known as lattice multiplication, which

divides the process into smaller steps while maintaining an algorithm that is precisely the same as the traditional long multiplication method.

Keywords: McEleice, Cloud, AWS, Cryptosystem, NTRU, Security

1 Introduction

When two ends of clients and web services are involved, WSS offers message-level protection. The security of the WSS architecture proposed in this work will be based on the Data Encryption Standard (DES). It simulates DES-based web service security and assesses the WSS's effectiveness using certain common criteria. Experiments were carried out utilizing the appropriate tools in a Windows Vista operating system environment to enable system evaluation. The evaluation's findings showed that implementing security alongside web services entails additional costs, SCTPWE, and Request/Respond Time with Encryption that rises as a function of both (TRRWE) [1]. Elliptic-Curve Diffie-Hellman (ECCDH) was used to make sure that the process of initiating a prime number which is in range, cyphertext, and the public key of the device are comprehended by the intended machine [2]. The combination of cryptography and steganography, or cryptography, may be a useful "tool" for boosting the security of cloud-based computations and/or communications. This paper thus suggests a new cryptographic system that combines steganography and post-quantum cryptography to guarantee that cloud communication security is preserved both in the era of classical computing in addition to the Post-Quantum Computing era [3]. The cloud computing architecture is different from other delivery methods for computer resources and services. Cloud computing architecture has the potential of improving healthcare. Before it is extensively used, cloud computing should be carefully studied as with an invention. The main aim of this article is to evaluate the merits and demerits based on the concept in context of administration, technology and law. [4]. The discrete logarithm and integer factorization issues serve as the foundation for the majority of currently used conventional cryptographic methods. While utilising classical computers, these tasks are regarded as being difficult, quantum computers have been demonstrated to offer significant speedups using techniques like Shor's algorithm to circumvent such systems. To provide cryptographic primitives immune to quantum assaults.Many contemporary algorithms, including cryptographic methods based on lattices and codes, are believed to be safe in the presence of quantum computers. With various modifications, the code-based The McEliece cryptosystem has been in use for 30 years and continues to be used today. Two McEliece-based algorithms have been developed as potential standardization candidates for digital signature and encryption techniques that are resistant to quantum mechanics [5]. The paper proposes a methodology for securing access control data using a form of the McEliece cryptosystem, which is expected to displace RSA in the future. Cloud user data is secured using a variation of the NTRU. When the new NTRU algorithm was simulated its was revealed that the time complexity had been enhanced even further[6]. Service Level Agreement (SLA), a legal document made available to

consumers by cloud developers, is considerably complied with by job scheduling and load balancing. Key SLA requirements like Deadline are considered by the LB algorithm, the priorities of VMs and amount of the resources that are to be allocated by optimizing the resources. Recommended LB algorithm addresses both the issues that have been identified and the area of study that is currently lacking. Additionally, it does well in terms of quicker make-up and execution times [7]. An idea for a cloud computing data security model is presented in this work. There are defences against several common assaults, like SQL injection and brute force. Due to the crucial role that data encryption plays in data security, a number of contemporary encryption approaches are implemented and assessed to determine the best method to employ in a cloud environment. Using NIST statistical testing, the following encryption methods—RC4, DES, AES, Blowfish, RC2, and 3DES—are assessed for randomness. The amount of time needed for each method to decrypt the same amount of data is also taken into account. [8]. A paradigm that would provide sufficient data security for both conventional and quantum computing is urgently needed, as encryption is now the most prevalent way of cloud data protection. [9]. The McEliece algorithm's simulation showed that it was temporally less complex than the current one. [10]. Only the numbers that are prime are made use of in the changed RSA algorithm. We employed the difficult-to-crack prime number "n" in our method[11]. When integrated with the DSBT method, the POR algorithm's computational efficiency is improved, which has no bearing on file size and can reach constant-level calculation complexity. We use refinement and data segmentations to enhance the access control technique, verification of label of data and Cloud integrity[12]. Multiple ways of facing the issue of quantum security are available and as of now the some of the algorithms include code-based, multivariate, supersingular elliptic curve and lattice based. All these algorithms gave pros and cons of their own based on the performance[13]. Since cloud computing was developed and adopted widely, data security has been a major issue. The article suggests a way for protecting cloud user data with an NTRU cryptosystem and access control data with a McEliece algorithm variation.It was shown that the new NTRU algorithm outperformed the currently employed one by modelling the NTRU approach that was proposed[14]. In order to increase the security of the NTRU cryptosystem by making it resistant to assaults from both conventional and quantum computing, a variation of the cryptosystem is created[15]. As soon as the user's role was established, the associated permissions were fixed, and the security control strength was altered at the system's real-time congestion level. The same position has several activation permissions under the management of different security strengths. [16]. Lattice-based public key cryptography is used in NTRU. In NTRU, the processes for encrypting and decrypting data are based on polynomial multiplication. Due to this characteristic, NTRU is much faster than other public key encryption algorithms like elliptic curve cryptography and RSA. Hardware implementation of NTRU using the residue number system is described in order to speed up implementation. The encryption and a portion of the decryption process are done by taking into consideration RNS bases in order to achieve high implementation speed. The outcome demonstrates a discernible improvement over the original NTRU cryptography[17]. By developing a method that uses the cryptographic methods AES and SHA-2 in conjunction, have enhanced the security of data. A secure cloud storage

solution for SMEs was developed and implemented. A method for ensuring data integrity and secrecy when delivering data in the cloud AES was developed by M. Meenakumari and G. Athisha by integrating the hash function (MD5) with the encryption algorithm. [18]. Since they only need a single private key at both ends for encryption and decryption, symmetric key cryptosystems are, in comparison, quicker than public key cryptosystems. Even yet, it might be challenging to maintain security in a hostile environment while carrying out compatible and efficient key distribution and safe private data transfer across entities. Public keys are used in the 1978-created McEliece cryptosystem, which also depends on unnamed error-correcting codes for security. This paper details the component-by-component strategy utilised in the development of this cryptosystem and provides a state-of-the-art complete analysis of it. Separate sections are devoted to each of the several McEliece cryptosystem attacks. The experimental findings utilising Goppa codes are also presented in the paper where the simulations are done on various extension degrees. We came at the following findings as a result of our simulations[19]. Provided security architecture provides greater data confidentiality and security. In the new security architecture, blocks of bits are used to separate data. For each couple of blocks of bits, the genetic algorithm is used. Each encrypted data is stored at a separate area in the cloud, and its precise position is unknown. It is therefore challenging for an attacker to understand the message. Additionally, genetic algorithms lack a fundamental idea that promotes data security. The new security architecture employs evolutionary algorithms on smaller blocks to increase security. Another tool the framework uses for secure and precise data access is the capability list[20]. a novel hybrid approach, where the Advanced Encryption Standard (QAES) algorithm and the QKD algorithm are implemented as the first security methods for cryptography functions using techniques that generate keys. This is known as one of the first hybrid technique in cloud computing, and it solves the short-range QKD problems and the Key availability issues of the QAES. [21]. NIST released a request for suggestions in 1997 for an improved AES that should be more efficient and secure than 3DES. Some of the necessary criteria included security, computational effectiveness, memory needs, suitability of hardware and software, and flexibility[22]. The most challenging issue with cloud computing, a young technology, is security. If the vale of the public key is known to the hacker, the already existing RSA algorithm might not be suitable to protect the data as it is being sent across the network. The hybrid RSA encryption method put forward in this paper makes it difficult for an adversary to decipher the data.The proposed method gets over the first constraint by boosting security without noticeably slowing down processing [23]. To address this problem and offer the first execution of such a strategy. The parties in our system, on the other hand, declare an associated access structure over attributes when they encrypt a message. In order for a user to be able to decode the data, certain user attributes must pass through the ciphertext's access structure. A tree, also known as access tree has the leaves and nodes in which the nodes indicate the threshold of the gate and the leaves indicate specifications [24].

2 Literature survey

The development of new technologies like utility computing, grid computing, and distributed computing has accelerated the tendency towards cloud computing. The most widely used computing technology today is being driven by cloud computing, which has shown great promise in providing adaptable, affordable, and powerful tools over the Internet. Virtual computers use network tools to access data while the cloud offers remote access and data storage. Cloud computing is an additional force behind the Fourth Industrial Revolution. When we use Microsoft Office 365, Dropbox, and other Google services, we are all using the cloud in our daily activities. While such an environment has many benefits, it also raises security concerns regarding data availability, cyberattacks, access control, privacy, and security as well as performance and reliability problems. Cloud service providers must put effective security and privacy means in place to guarantee data privacy, confidentiality, integrity, and availability. However, end customers do not receive enough dependable and secure services from cloud service providers. Blockchain is a tool that improves cloud processing. This innovative technology provides strong data fidelity capabilities while addressing security concerns. The paper offers a thorough review of the problems relating to cloud security and privacy. It serves as an example of the importance of security concerns. As a result, researchers will be inspired to investigate cloud computing security issues in the future. [1]. We use McEliece algorithm for the decryption and encryption of user creds, while the NTRU algorithms is used for the decryption and encryption of user data. When a cloud administrator forces users to encrypt and decode their user credentials, the McEliece cryptosystem is advised. For authentication purposes, users who want access to the cloud must enter their user credentials. Users obtain their credentials from the cloud administrator, it should be noted. In employing a new NTRU cryptosystem, which is under consideration uses lattice arithmetic to encrypt and decode data. For multiplying larger numbers or performing complex multiplication, use the lattice approach. Although it divides the procedure into smaller phases, the algorithm is the same as the conventional long multiplication approach. The suggested NTRU cryptosystem has to do the following actions. [2]. There are certain difficulties notwithstanding much prior study in the field of cloud computing. Cloud-based apps are nevertheless taken into consideration, particularly when it comes to workload balancing in infrastructure. model for cloud IaaS. Effective work distribution constitutes critical component of cloud computing. Limited resources and virtual machines. One paradigm for this technology is IaaS. a management system for servers, data centers, and virtual machines. services for the cloud. High serving performance is ensured by such a paradigm, which also prevents issues like host overload. Or it is not used enough since it causes longer execution times, system downtime, etc. Task scheduling is generally in compliance with service level criteria, and it helps with load balancing. (SLA) Agreement, an item that the user receives from the cloud developer. The LB algorithm handles key SLA criteria like deadlines. The suggested technique optimises task parameters, VM priority, resource allocation, and Quality of Service (QoS) metrics. The LB algorithm under consideration handles the aforementioned problems and open research questions. drawn on research in the literature. Comparing the suggested LB algorithm to current

dynamic LBA algorithms, the findings revealed that the proposed algorithm produces an average resource utilisation of 78D44. Furthermore, it performs admirably. With respect to execution speed and makespan reduction [3].Several privacy and security-increasing techniques are invented to ensure proper security of the internet. Users should be able to converse anonymously while forwarding emails, browsing the net, or sending messages to groups by employing these techniques. In order to expand the number of people who could participate in their secure communication, we developed a method for anonymous contact. It safeguards the relationship between output and input[4]. The need for distribution grows as technology progresses and new Internet gadgets are developed. measures made to safeguard the privacy and integrity of personal and business data. any host gadget with internet access. Systems that impose boundaries between computers are known as security systems. Network (Bishop, 2003). (Bishop, 2003). The security of computer-related systems and more internet-enabled gadgets has occasionally been needed. Hackers and crackers simply exist, and everyone plays along. Companies that produce or use computers work to improve their internet security. Successful. Attacks on computer systems and the information they contain can jeopardise their dependability, availability, and confidentiality. Current and future security concerns poses a great threat to the management of security and internet information. DoS/DDoS cannot be entirely avoided, but it can be minimized with appropriate measures. Block phishing addresses or unauthorized access from the host device's viewpoint. Eliminating this service is essentially impossible, so working with a host server is preferable. Attacks called "man-in-the-middle" happen when a user is eavesdropping on data being sent between a sender and a recipient. Replay attacks, according to Microsoft (2017), involve an attacker sending a message stream between her two partners and to one or more of them. Otherwise, It is impossible to prevent repetition of elements because this is regarded as a legitimate message from a computer [5]. All companies today, as well as everyday people, are using the cloud for their processing and storage requirements. There are many advantages to using internet storage, and all cloud service providers store data redundantly. Customers of your business can work together using cloud computing by saving data there. Data is encrypted using a 256-bit password and 14 computation cycles in this specific 256-bit variant. However, there are some drawbacks to the aforementioned application. Any encryption technique is attackable if the ciphertext can be decoded before using brute force. Attacks on implementation, also referred to as side-channel and key recovery attacks, commonly target AES. In as little as one-third to five percent of the actual time, AES's tried-and-true key recovery method can locate the algorithm's key. The assault was made possible by the AES all-round change design's resistance to various attack groups [6]. Performing compatible, effective key distribution and safe private data transmission between organisations in an untrusted setting remain security challenges. This white paper gives a current, comprehensive summary of this cryptosystem, along with component-by-component explanations and implementations of its algorithms. Different dangers to the McEliece cryptosystem are covered separately. The Goppa code experiment findings are also included in this article. Here, different development rates are used in the algorithms. We listed several implementation-related issues along with the results of the models we performed. [7].

3 Methodology

3.1 Existing System

- ABE, Simple shared key, Multi entry key and tree base key managements are some of the possible management models that are used in the current approach.
- Each client who has enrolled for the relevant group is given a key under attribute-based encryption (ABE), which has both advantages and disadvantages. The key assignment and computational cost of this system are its major flaws.
- Control for multiple input keys The primary flaw, after thwarting side channel and inactive attacks and enhancing identity management, is that SLA conflicts increased the likelihood of sniffing attacks.
- While SSK 1 client side does prevent attacks such as DoS and active attacks, it does however have a demerit that is that its key is weak against attacks that are for authentication.
- Tree-Based key management designs a secure and flexible key management mechanism. the major disadvantage leakage for key generation and distribution.

3.2 Demerits of Existing System

- Vulnerability to brute-force attacks: Some encryption methods rely on a relatively small key size, which makes them susceptible to brute-force attacks. If an attacker can try enough combinations of keys, they may be able to crack the encryption.
- Vulnerability to known plaintext attacks: Some encryption methods are vulnerable to attacks in which an attacker has access to both the plaintext and the ciphertext, allowing them to deduce the key used to encrypt the data.
- Vulnerability-chosen plaintext attacks: In a chosen plaintext attack, an attacker can select the plaintxt that is encrypted and then use that knowledge to deduce the key used in the encryption.
- Vulnerability to Side-channel attacks exploit weaknesses in the physical implementation of the encryption algorithm rather attacking the algorithm itself. For eg, an attacker might be able to deduce the key used in an encryption by analyzing the power consumption of the device doing the encryption.
- Key distribution: Many encryption methods rely on securely distributing the key used in the encryption to all parties
- who need it. This can be a difficult problem in practice, especially when the parties are not physically co-located.
- Quantum computing: Some encryption methods, such as RSA, are vulnerable to attacks by quantum computers, which could potentially break the encryption in a much shorter time than a classical computer.
- Backdoors: There is a risk that encryption methods may include backdoors, either intentionally or unintentionally, that could be exploited by attackers. This has been a concern with some widely used encryption methods in the past.

3.3 Problem statement

To ensure the confidentiality and privacy of personal data while it is being transmitted and stored. In the current digital age, data is constantly being moved across networks and stored in a variety of devices, making it vulnerable to unauthorised access, theft, and tampering. The solution to this problem is encryption, which converts plaintext data into ciphertext using an algorithm and a key, making it unintelligible to those without access to the secret. Decryption is the process of transforming the ciphertext back into plaintext while maintaining the same method and secret. The challenge is to implement robust encryption and decryption methods that are resistant to attacks by hackers and other malicious actors. Another challenge is to balance security with usability, ensuring that encryption and decryption do not hinder the ability of authorized users to access and use data.

3.4 Advantages of Proposed System

- The first benefit is that encapsulation and decapsulation happen quickly. The uniformly random polynomial a1 and another polynomial a2 = a1s1 + s2, where si are kept secret, make up the public key of Ring-LWE-based primitives.
- To make the public key smaller, one creates a1 = H(k), where k is a brief random number, and only keeps k, a2 as the public key.
- If H is an XOF based on SHAKE or AES, then computing a1 = H(k) during encapsulation and decapsulation may be a rather expensive process (in contrast to all the other computations involved).
- Public keys and ciphertexts based on NTRU may result in quicker and more compact primitives that make use of zero-knowledge proofs.

3.5 Proposed System

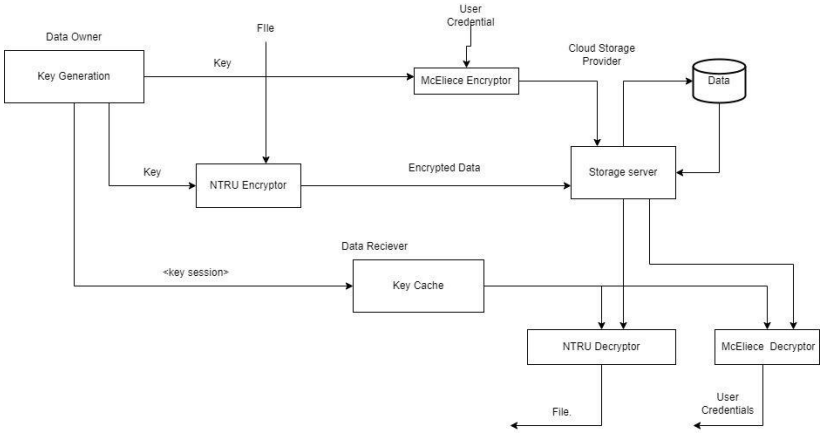

Fig.1 Illustrates the **system architecture** that the hybrid algorithms follow in order to safely encrypt and decrypt user credentials and data using NTRU and McEliece algorithm

Fig1. Illustrates the system architecture of the Hybrid McEliece and NTRU cryptosystem. Here it can be observed that first the keys, both private and public are generated and then the private key is used by McEliece and NTRU algorithms for the encryotion of user credentials by the McEliece algorithm and then the NTRU algorithm is used for the encryption of user's data. Both the McEliece and NTRU algorithms are used together to get a good layer of security. Then the public keys are uploaded to the cloud and then they can be accessed by the user whenever they want to access their user data. All they have to do is firstly, decipher the ciphertext using the public keys obtained by the user on the cloud. Then the user credentials can be deciphered and by using the user credentials the user can access the user data. For that the public key is also available, after obtaining the public key the user data can be deciphered using NTRU/McEliece decryptor.

3.5.1 McEliece Cryptosystem

The McEliece algorithm is being used in this article to ensure the safety of the user credentials of the user that are uploading their precious data on the cloud. Basis of the algorithm is the difficulty that is carried to decode the codes which are linear. A code is designed that is capable of self-correcting for which an appropriate algorithm is chosen. Goppa codes(binary) are used in the authentic algorithm. An algorithm known as Patterson, has made it very easy to decipher these codes. In order to make the public key, the private key is modified using some code as code that is linear. Cryptanalysis has been futile thus far. But the encryption times of this algorithm are faster than RSA and some other algorithms making it a better option.

P: permutation matrix, *G*: Goppa codes

S: non-singular matrix, *a*: message that needs to be encrypted

For key generation:

The key is generated by the formula given below:

$$A_{key}=S_T \cdot A \cdot P_T$$

Where transpose is represented by T

For Encryption:

For encryption of the message, below step is followed:

$$C^I=m \times A_{key}$$

$$C=C^I+z$$

For decryption:

For decryption of the message, below step is followed:

$$d=C(P^T)^{-1}$$
$$m=X(S^T)^{-1}$$

And the encrypted user credentials are decrypted.

3.5.2 NTRU Cryptosystem

The NTRU encryption algorithm is used in this article to propose a novel technique for safeguarding files kept in the cloud. Based on certain polynomial rings' mathematical properties, the NTRU was developed. Shortest vector in a lattice is the problem that the NTRU is was developed for. NTRU will have two keys: a private key and a public key. The private key will be used by this algorithm at the time of decrypting the data NTRU methods are applied over a ring of truncated polynomials. To encrypt and decode the user credentials, the McEliece algorithm will be applied. To the cloud administrator will be relayed the details of the encrypted passwords.

- N is the number of polynomials with degree N-1 in the ring R.
- P and Q, which stand for small and big moduli, are used to reduce the coefficients in data encryption and decryption.
- The public key is handled by the polynomials z and v.

For generating the key:

Sender calculates z·z t = 1.(mod t) and z·z y = 1(mod y) and then creates the public key k:

k=tzy·v(mody)

For Encryption:

For encryption of the message, below step is followed:

x=j·k+m(mody).

For Decryption:

For the decryption of the message, the step below is followed:

n=z·x(mody),n=z·x(mody),

l=n(modt),l=n(modt),

C=zt·l, C=zt·l,

This translates to the message that was initially encrypted.

4 Implementation

NTRU a public-key encryption algorithm that can be used to encrypt user data. The process of cryptography consists of various steps including key generation, encryption, decryption etc. NTRU uses polynomial for the encryption of data. The McEliece cryptosystem is a public-key encryption algorithm that is based on the hardness of decoding a general linear code, which is known to be an NP-hard problem. NTRU and McEliece are being used in conjunction with each other. McEliece is being used for the encryption of user credentials such as user name and user password while the NTRU algorithm is being used for the actual encryption and decryption of user data. First we create a key using McEliece algorithm, a McEliece public key and associated private key must is created. While the private key is kept confidential and is uploaded to cloud and later used for decryption, the public key is used for encryption. Typically, a trusted key generation method that adheres to the McEliece criteria is used to generate the key pair.User credential is created as a plaintext and after that it is encrypted. The plaintext is then converted to a message vector, which is a binary vector of fixed length that represents the plaintext. This is done using by converting the plaintext to its corresponding binary representation. The message vector is encrypted using the McEliece public key and the McEliece encryption algorithm. This is done by adding a random error vector to the message vector, and then multiplying the resulting vector with a randomly generated matrix.The result of the McEliece encryption process is a ciphertext, which is the encrypted representation of the message vector.After the user credentials have been encrypted and the public key and private key has been shared to the user and the cloud respectively, next comes the process of encrypting the user data.The process of encryption of user data is similar to the process using McEliece algorithm. The user data to be encrypted is first prepared as plaintext, which can be a message, a file, or any other data. The plaintext is then converted to a polynomial representation by converting the plaintext to its corresponding numerical values and representing them as polynomial coefficients. This typically involves multiplying the plaintext polynomial with a random polynomial, adding some random noise, and applying modular arithmetic

operations. The result of the NTRU encryption process is a ciphertext, which is the encrypted representation of the plaintext. This encrypted ciphertext is stored in the cloud provided by AWS.

The next step involves the decryption of user data by the intended user. This is done in the following way:

The ciphertext is decrypted using the McEliece private key and the McEliece decryption algorithm. This is done by applying matrix operations to the ciphertext, and then correcting the errors introduced during encryption using the syndrome decoding technique. The decrypted message vector is then converted back to its original representation as plaintext, using the inverse of the method used during encryption. And thus he user obtains the user credentials required for accessing the encrypted user data. The ciphertext is decrypted using the NTRU private key and the NTRU decryption algorithm. This is done by applying modular arithmetic operations and polynomial arithmetic to obtain the original plaintext polynomial. The plaintext polynomial is then converted back to its original representation as user data, using the inverse of the method used during encryption. This involves converting the polynomial coefficients to numerical values and interpreting them as the original data. Hence, this is how the encryption of user credentials and user data is done using McEliece and NTRU algorithm. The security of the McEliece algorithm relies on the appropriate choice of parameters, such as the length of the message vector, the size of the error vector, and the properties of the randomly generated matrices. Proper implementation and management of the McEliece key pair, including the generation, storage, and protection of the private key, is also crucial for ensuring the security of user data encrypted using McEliece. It's important to note that the security of NTRU encryption relies on the appropriate choice of parameters, such as the degree of the polynomial, the modulus, and the random polynomials used during encryption. Proper implementation and management of the NTRU key pair, including the generation, storage, and protection of the private key, is also crucial for ensuring the security of user data encrypted using NTRU.

5 Result analysis

filesize (kb)	AES(s)	RSA(s)	QAES (s)	Old McEliece (s)	New-McEliece (s)
21	0.6620	0.19	0.0100	17.6008	14.5679
78	0.8643	1.22	0.0452	64.4509	56.5589
153	0.8777	1.52	0.0775	141.9506	130.8998
283	0.9426	1.92	0.0882	250.9907	240.7236
305	1.2863	3.51	0.0933	268.8508	250.6215
through-put	2342.15	15452.561	322.1212	3.755	2.2531
av-eragetime	0.9452	2.22	0.0891	148.7489	140.8459

Table 1: Comparison of the proposed and existing McEliece algorithms' execution times.

Table 1 shows time taken for the encryption and decryption of data by using various algorithm is given. McEliece algorithm is used here for the encryption and decryption of the user credentials such as the encryption and decryption of user name and user data. Previous algorithms did not have the functionality of encrypting and decrypting user credentials. Encrypting the user credentials provides an extra layer of security to the data. If an attack were to happen, the attacker would first have to obtain the user credentials which have already been secured using the algorithm. QAES is an encryption algorithm that requires the aid of supercomputers and can be broken if someone has an access to supercomputer. But this algorithm doesn't require any supercomputer for encryption and can be done on a normal workstation.

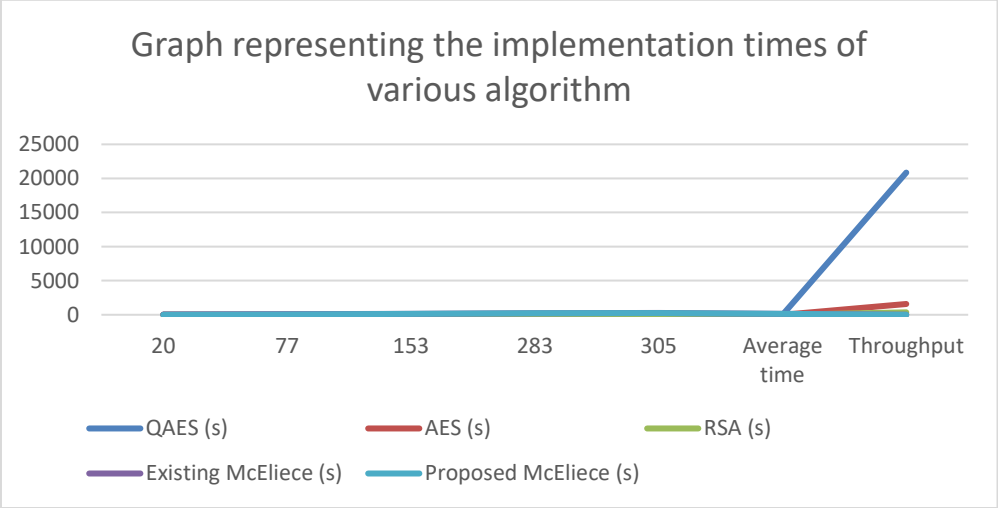

Fig. 2: The above figure illustrates the encryption time of some of the existing algorithms and a comparison is drawn with the new one

Fig. 2 we can see, the throughput values and average times taken for the encryption and decryption of user credentials by various algorithms are given. It gives insight about the way various algorithms are used for the encryption and decryption of user credentials and which one is more secure and efficient. This helps in better understanding the

efficiency of the algorithm and which one must be used for best results. Here we can see that the highest average throughput and time is taken by the Quantum AES algorithm followed by AES and McEliece.

Filesize (kB)	AES(s)	RSA (s)	QAES (s)	Existing NTRU (s)	Proposed NTRU (s)
22	0.6520	0.17	0.0100	811.0032	1980.126
75	0.8543	1.22	0.0452	3243.864	8076.654
153	0.8777	1.52	0.0775	6560.97	18856.66
283	0.9426	1.92	0.0882	12224.23	44385.51
305	1.2863	3.51	0.0933	37438.79	96772.25
throughput	2342.15	15452.561	322.1212	12055.77	33589.51
typical execution time	35623.5	1629.52	630.69	0.07	0.01

Table 2: Comparison between the execution times of old NTRU and Proposed NTRU Algorithm.

Table 2 shows a comparison of existing and proposed NTRU algorithm, with some older algorithm such as QAES, AES and RSA. While all these other algorithms use bit keys for the encryption and decryption of user data, the proposed NTRU algortihm will use polynomial based keys. The proposed NTRU algorithm is supposed to be very hard to crack. The reason behind is not that it is a very secure algorithm but it will take such a long amount of time for the encrypted data to be decrytped that by the time it is decrypted it will have become useless. The strength of the NTRU algorithm live within its durablity against the attackers.

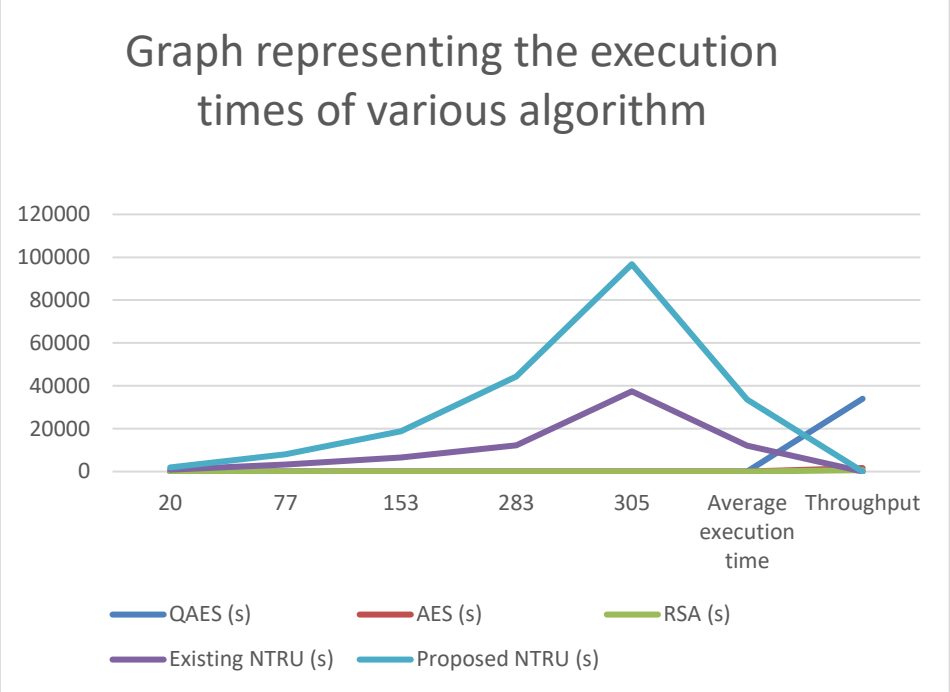

Fig.3: The above figure depicts the differences between the encryption time between the old algorithms and the NTRU algorithm

Fig. 3 we can see, the throughput values and average times taken for the encryption and decryption of user credentials by various algorithms are given. It gives insight about the way various algorithms are used for the encryption and decryption of user data and which one is more secure and efficient. This helps in better understanding the efficiency of the algorithm and which one must be used for best results. Here we can see that the highest average throughput and time is taken by the Quantum AES algorithm followed by AES and NTRU.

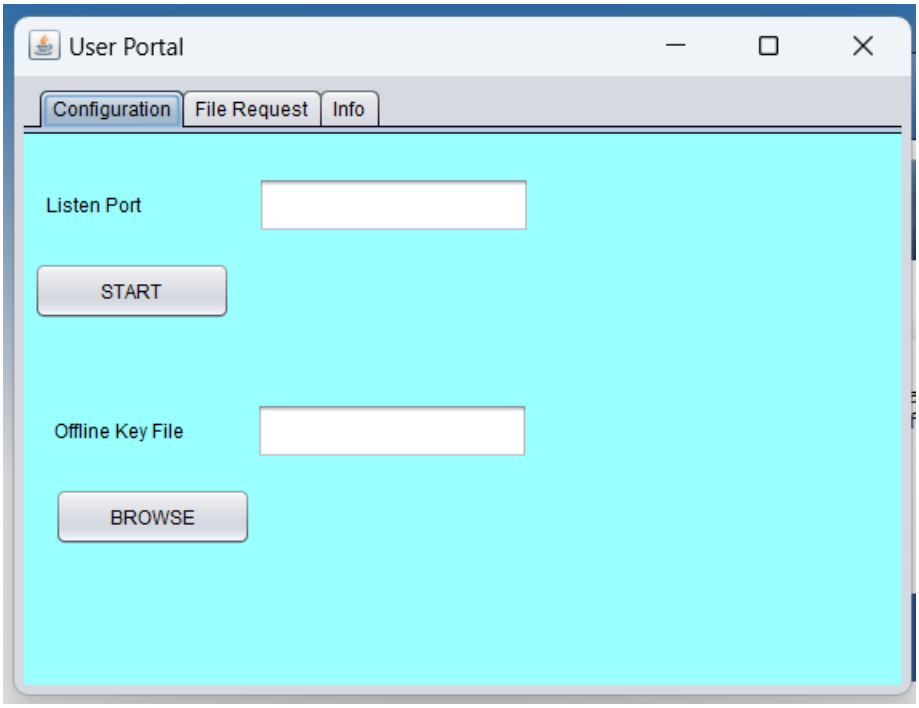

Fig.4: Illustrates how the user portal will look like and the option to request the decrypted file along with the information.

Fig 4. it is here that the user can request and download the file that has been encrypted, and the decrypted file will be downloaded. All the user needs to do is put the right listening port and start the transmission and then request the file that they need. The name of the requested file should be the same as the name of the encrypted file.

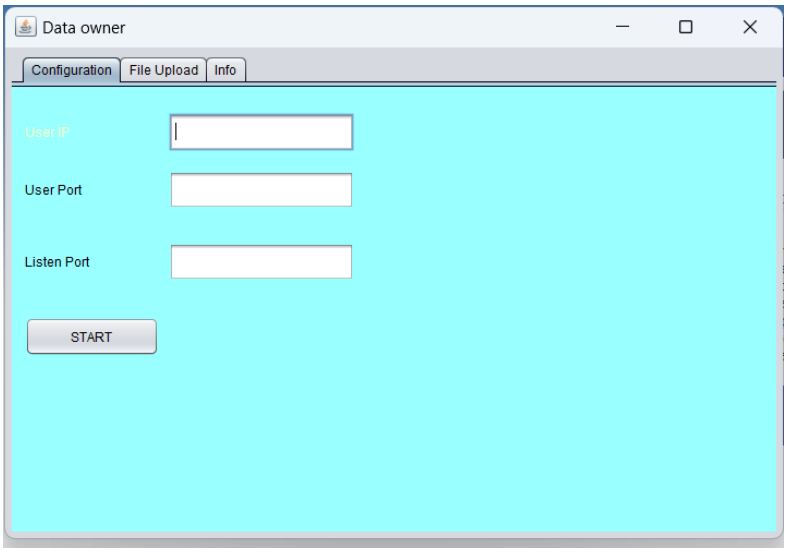

Fig. 5: The above figure illustrates the data owner portal that is used to upload the file which will then be encrypted.

Fig. 5 This is the data owner portal. It is here where the file is encrypted. The user needs to first initiate the connection by giving proper user IP and user port. Once the user port has been initiated, in the file upload section the file needs to uploaded. The file that is uploaded will be the encrypted file and it will be uploaded to the AWS server in five equally encrypted txt files. The user can then request the file and the decrypted file will be sent to the user almost immediately.

6 Conclusion and Future work

NTRU and McEliece are all important cryptographic algorithms that have been studied extensively in recent years. It has several advantages over other public-key cryptosystems, including its resistance to quantum computing attacks and its relatively fast encryption and decryption times. However, it also has some weaknesses, such as its vulnerability to certain attacks, particularly those based on lattice reduction algorithms. While it is still in the experimental stage, it shows promise as a potential post-quantum cryptographic algorithm. However, it has some drawbacks, including its large key sizes and slow encryption and decryption times. In conclusion, all three of these cryptographic algorithms have their strengths and weaknesses, and further research is needed to improve their efficiency, security, and applicability in real-world scenarios. In the future, there will likely be continued interest and development in these and other post-quantum cryptographic algorithms as the threat of quantum computing attacks continues to grow.

For NTRU, future work could focus on further improving its resistance to lattice-based attacks while also reducing its computational requirements. Research could also explore the potential for hybrid cryptosystems that combine NTRU with other post-quantum algorithms for enhanced security For McEliece, future work could focus on reducing its key sizes and improving its encryption and decryption times while maintaining its resistance to attacks by both classical and quantum computers. Research could also explore the potential for hybrid cryptosystems that combine McEliece with other post-quantum algorithms for enhanced security. In general, all three algorithms will require ongoing research and development as the field of post-quantum cryptography continues to evolve. This includes testing and analysis of their security properties, as well as exploration of new algorithmic approaches and optimizations. Additionally, efforts to standardize and integrate these algorithms into existing cryptographic systems will be important for their adoption and deployment in real-world applications. This cryptosystem can prove to have a great deal of potential in various institutions that employ the need of encryption and decryption. For example, banks, grocery stores, healthcare, schools, universities and a plethora of government official are some of the institutions that require to have utmost security because they hold some of the most sensitive information about individuals.

References

1. “Security and privacy aspects of cloud computing: a smart campus case study” M. A. Razzaq, M. Ahmad, S. H. Gill , N. Z. Jhanjhi, F. M. Almansour, I. Haq, et al., vol. 31, no. 1, pp. 117–128, 2022, 10.32604/iasc.2022.
2. HC Ukwunoma, G Arome, A Thompson “Post-quantum cryptography-driven security framework for cloud computing” doi.org/10.1515/comp-2022-0235
3. “A load balancing algorithm for the data centres to optimise cloud computing applications,”] D. A. Shafiq, N. Z. Jhanjhi, A. Abdullah, and M. A. Alzain, vol. 9, pp. 41731–44, 2021, 10.1109/ACCESS.2021.3065308.
4. Ahmad, Khaleel, Kamal, Afsar, Bin Ahmad Fast hybrid- MixNet for security and privacy using NTRU algorithm doi.org/10.1016/j.jisa.2021.102872
5. “A secured system of Internet Enabled Host Devices,” D. A. Shafiq, N. Z. Jhanjhi, A. Abdullah, and M. A. Alzain,, vol. 5, no. 1. pp. 26–36, 2020.10.5539/nct.v5n1p26
6. N. Rani, N. Juliet, and S. Arunkumar, “A novel cryptosystem for files stored in cloud using NTRU encryption algorithm,” Int J Recent Technol Eng (IJRTE), vol. 9, no. 1. pp. 2277–3878, 2020.10.35940/ijrte.A2536.059120
7. R. Kumar, A. S. Naidu, A. Singh, and A. N. Tentu, “McEliece cryptosystem: simulation and security vulnerabilities,” Int J Comput Sci Mathematics, vol. 12, no. No 1. pp. 64–81, 2020.10.1504/IJCSM.2020.108787
8. H. Shukur, S. Zeebaree, R. Zebari, D. Zeebaree, O. Ahmed and A. Salih, "Cloud computing virtualization of resources allocation for distributed systems", *J. Appl. Sci. Technol. Trends*, vol. 1, no. 3, pp. 98-105, Jun. 2020.
9. S. K. Mishra, B. Sahoo and P. P. Parida, "Load balancing in cloud computing: A big picture", *J. King Saud Univ.–Comput. Inf. Sci.*, vol. 32, no. 2, pp. 149-158, 2020.
10. Chowdary, P.R.; Challa, Y.; Jitendra, M.S.N.V. Identification of MITM Attack by Utilizing Artificial Intelligence Mechanism in Cloud Environments. *J. Physics Conf. Ser.* **2019**, *1228*, 012044.

11. D. A. Shafiq, N. Jhanjhi and A. Abdullah, "Proposing a load balancing algorithm for the optimization of cloud computing applications", *Proc. 13th Int. Conf. Math. Actuarial Sci. Comput. Sci. Statist. (MACS)*, pp. 1-6, Dec. 2019
12. Anand, S.; Perumal, V. EECDH to prevent MITM attack in cloud computing. *Digit. Commun. Netw.* **2019**, *5*, 276–287.
13. Huang, Q.; Yang, Y.; Yue, W.; He, Y. Secure Data Group Sharing and Conditional Dissemination with Multi-Owner in Cloud Computing. *IEEE Trans. Cloud Comput.* **2019**
14. Miao, Y.; Liu, X.; Choo, K.-K.R.; Deng, R.H.; Li, J.; Li, H.; Ma, J. Privacy-Preserving Attribute-Based Keyword Search in Shared Multi-owner Setting. *IEEE Trans. Dependable Secur. Comput.* **2019**
15. G.S. Mahmood, J. H. Dong, and B. A. rahman Jaleel, "Achieving an effective, confidentiality and integrity of data in cloud computing," *International Journal of Network Security*, vol. 21, no. 2, pp. 326–332, 2019.
16. K. V. Pradeep, V. Vijayakumar, and V. Subramaniyaswamy, "An efficient framework for sharing a file in a secure manner using asymmetric key distribution management in cloud environment," *Journal of Computer Networks and Communications*, vol. 2019, Article ID 9852472, 8 pages, 2019.
17. O. C. Abikoye, A. D. Haruna, A. Abubakar, N. O. Akande, and E. O. Asani, "Modified advanced encryption standard algorithm for information security," *Symmetry*, vol. 11, no. 12, p. 1484, 2019.
18. S. Nur Rachmat, "Performance analysis of 256-bit AES encryption algorithm on android smart phone," *IOP Conf. Series: Journal of Physics: Conf. Series*, vol. 1196, 2019.
19. H. A. Al Essa and A. S. Ashoor, "Enhancing performance of AES algorithm using concurrency and multithreading," *ARPN Journal of Engineering and Applied Sciences*, vol. 14, no. 11, 2019.
20. S. Othman and A. S. Riaz, "A user-based trust model for cloud computing environment," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 3, 2018.
21. Dr. Ramalingam Sugumar and K. Arul Marie Joycee, "FEDSACE: a framework for enhanced user data security algorithms in cloud computing environment," *International Journal on Future Revolution in Computer Science & Communication Engineering*, vol. 4, no. 3, 2018.
22. K.-L. Tsai, Y.-L. Huang, F.-Y. Leu, I. You, Y.-L. Huang, and C.-H. Tsai, "AES-128 based secure low power communication for LoRaWAN IoT environments," *IEEE Access*, vol. 6, pp. 45325–45334, 2018.
23. S. Mall, and K. Saroj, "A new security framework for cloud data", 8th International Conference on Advances in Computing and Communication (ICACC), 2018.
24. "Cloud computing: A paradigm shift in the way of computing", D. A. Shafiq, N. Z. Jhanjhi, A. Abdullah, and M. A. Alzain, *Int. J. Mod. Educ. Comput. Sci.*, vol. 9, no. 12, pp. 38-48, Dec. 2017.