Research Article

Henry Chima Ukwuoma*, Gabriel Arome, Aderonke Thompson, and Boniface Kayode Alese

Post-quantum cryptography-driven security framework for cloud computing

https://doi.org/10.1515/comp-2022-0235
received May 20, 2021; accepted February 17, 2022

Abstract: Data security in the cloud has been a major issue since the inception and adoption of cloud computing. Various frameworks have been proposed, and yet data breach prevails. With encryption being the dominant method of cloud data security, the advent of quantum computing implies an urgent need to proffer a model that will provide adequate data security for both classical and quantum computing. Thus, most cryptosystems will be rendered susceptible and obsolete, though some cryptosystems will stand the test of quantum computing. The article proposes a model that comprises the application of a variant of McEliece cryptosystem, which has been tipped to replace Rivest–Shamir–Adleman (RSA) in the quantum computing era to secure access control data and the application of a variant of N-th degree truncated polynomial ring units (NTRU) cryptosystem to secure cloud user data. The simulation of the proposed McEliece algorithm showed that the algorithm has a better time complexity than the existing McEliece cryptosystem. Furthermore, the novel tweaking of parameters S and P further improves the security of the proposed algorithms. More so, the simulation of the proposed NTRU algorithm revealed that the existing NTRU cryptosystem had a superior time complexity when juxtaposed with the proposed NTRU cryptosystem.

Keywords: Cryptography, Public-key cryptography, private key cryptography, data security, quantum cryptography

1 Introduction

With the emergence of cloud computing, the provision of data security through encryption has been a major technique to safeguard data against attackers/hackers [1]. Usually, data exchange is carried out in mediums that are not secure enough, which gives room for the interception of data by intruders [2–4]. According to ref. [4], cloud computing promises enhanced data security, reduced cost for services, enhanced flexibility, and higher availability, but the knowledge domain shows that frameworks for various data security models have been proffered for enhanced cloud data security using advanced encryption standard (AES), data encryption standard, RSA, and Elliptical Curve Cryptography (ECC), yet data security issues still prevail.

Cryptosystems such as RSA and ECC based on factoring problems and discrete logarithms, respectively, have sufficiently provided cloud data security for years over all forms of classical attacks. However, today's data-driven society is gradually drifting from classical computing, where information is stored in 0s and 1s to quantum computing, where information is stored in qubits. The advent of quantum processors and hence quantum computing, has revealed potential weakness in existing cryptosystems, thus necessitating the urgent need to source for alternatives that will ensure data protection.

Frequently used Cryptosystems, for example RSA and ECC have sufficiently provided cloud data security for years over all forms of classical attacks, yet the theft of data prevails. Hence, the urgent need to deploy quantum safe cryptosystems that are both safe for data processing in the classical and quantum space.

This article proposes a variant of the Code-based cryptosystems and Lattice-based cryptosystems. It seeks to develop a robust hybrid cloud data security framework with a view to understudy the cryptosystems as mentioned earlier and the designing of a variant of McEliece cum NTRU cryptosystems, respectively, in a hybrid architecture.

1.1 The concept of post-quantum cryptography

The dominance and recognition of the need for the use of public, key cryptography such as RSA and ECC demanded researchers to find a proficient way for unravelling the

* **Corresponding author: Henry Chima Ukwuoma,** Department of Cyber Security, Federal University of Technology, Akure, Ondo State, Nigeria, e-mail: henro317@gmail.com
Gabriel Arome, Aderonke Thompson, Boniface Kayode Alese: Department of Cyber Security, Federal University of Technology, Akure, Ondo State, Nigeria

factorisation problem and discrete logarithm problem. The unravelling of these hard-mathematical problems will thus provide a breakaway for RSA and ECC security. While researchers have tried to solve the problems with the use of classical computers, Peter Shor in 1994 used a quantum computer to develop and demonstrate an algorithm for efficient factorisation [5]. It is pertinent to note that with the advent of quantum computers, the present security infrastructure cum cryptosystems, where users and internet users rely, will be rendered obsolete and irrelevant [6].

Post-quantum cryptography is a branch of study whose sole aim is to update and provide security for the current cryptosystems with the presence of quantum computers [7]. Research in this branch of study entails studying cryptosystems that make use of the factorisation problem and discrete logarithm problem, and remain secured against the two problems even though the hacker/attacker is armed with quantum computing. However, the National Academies of Sciences, Engineering, and Medicines describes quantum computing as the usage of the quantum phenomena (entanglement and superposition) to carry out computation for solving computational problems such as the integer factorisation of the RSA.

1.2 Classes of quantum cryptography

There are four major classes of quantum cryptographic algorithms that resist quantum attacks [8]. These are:

a) **Code-based cryptosystems**: These categories of cryptosystems adopt the principle of extracting the initial bits of data transmitted over a channel by encoding the data in a specific structure, which may be recovered to a certain number of errors during transmission. Additional bits of data are added during the encoding of the data to be sent and then decrypted on reception if the specific information about the coding structure is known. An example of the code-based cryptosystem is the McEliece cryptosystem.

b) **Lattice-based cryptosystems:** Lattice-based cryptosystems are the foremost candidates for public-key post-quantum cryptography [9]. They use multidimensional lattices on solving the hardness of certain problems. An example of a lattice-based cryptosystems is the NTRU cryptosystem.

c) **Multivariate public key cryptosystem:** Shehata [6] described multivariate cryptosystems as one that uses random sets of quadratic equations, and the processing of the encryption or decryption uses these equations at particular points.

d) **Hash-based cryptography:** Is a cryptosystem that uses hash functions to guarantee the integrity of messages. An example is the Merkle's hash-tree public-key signature system.

A group of international academics from the National Institute of Standards and Technology (NIST) carried out a research to find a solution to the imminent threat of rendering the present cryptosystems obsolete on classical symmetric and asymmetric cryptosystems as regards the emerging quantum computation. Table 1 below presents their findings:

Table 1: Cryptosystems under quantum computation

S no.	Cryptosystems	Current status
1.	AES	Large key sizes needed
2.	SHA-2	Larger output needed
3.	SHA-3	Larger output needed
4.	RSA	Broken
5.	Deffie–Hellman key exchange	Broken
6.	Elliptic curve cartography	Broken
7.	Buchmann–Williams key exchange	Broken
8.	Algebraically homomorphic	Broken
9	McEliece	Not broken yet
10.	NTRU	Not broken yet

Source [8].

It can be deduced from Table 1 that the dominant cryptosystems currently in use, which is RSA and ECC, are not secure under quantum computing as they both rely on hard mathematical problems. Quantum computation has excellent speed ups, which can easily solve these hard-mathematical problems. AES cryptosystems are required to use larger key sizes.

1.3 The security of cloud computing

According to the Organisation of NIST, cloud computing is described as a service model that enables immediate, simple, and on-demand available network access to shared computing resources such as servers, networks, data, applications, and services [10].

There are four cloud computing models: hybrid, public, private, and community clouds. Furthermore, the model also depicts service delivery models, which include

Infrastructure-as-a-Service (IaaS), the Platform-as-a-Service (PaaS), and the Software-as-a-Service (SaaS) (Figure 1).

Though it is expected that computing will serve as a utility such as telephone, gas, water, and electricity, it comes with a major challenge: data security problems. Cloud user reception of cloud services can be hindered due to security and privacy issues. Information sourced from the knowledge domain also reveals that cloud users feel reluctant to fully adopt cloud services because of security and privacy issues. Ref. [11] describes data security as a means of securing digital data against unauthorised users/actions. Data sharing is carried out in unsecured channels, which is susceptible to interception. This has led to cloud providers and clients resorting to various means of data protection techniques, one of such techniques is cryptography.

Additionally, data security encompasses three attributes: confidentiality, integrity, and authentication. Confidentiality entails the protection of information and restricting it from unauthorised access. This is achieved by the application of cryptography. Integrity ensures that unauthorised persons are not able to change or manipulate data intended for a specific user and could be achieved by the use of cryptography. It is pertinent to note that data has value only if it is safe. Data which has been manipulated does not have any value and may cause financial waste, for example, data manipulation in which information about financial accounts is stored. Similarly, cryptography plays an important role in ensuring data integrity. Frequently used methods of data integrity protection contain information about data changes and hash checksums by which data integrity is verified. Availability simply refers that information be made consistently/readily accessible for authorised parties. Availability also involves properly maintaining hardware, technical infrastructure, and systems that hold and display the information.

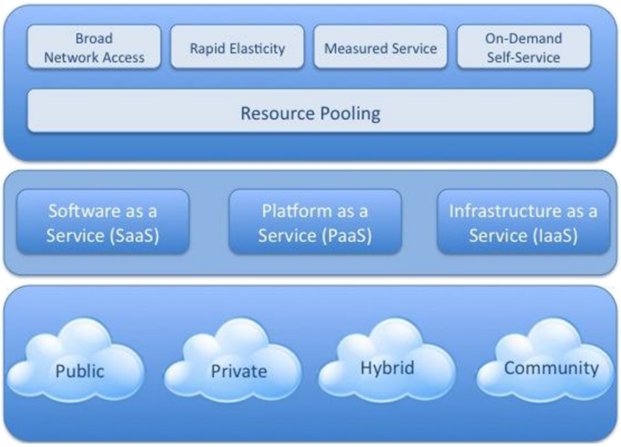

Figure 1: Cloud computing model [31].

2 Literature review

Ref. [12], proffered a secure framework for a multi-user and multi-owner cloud environment. The authors opined that security, integrity, and privacy of cloud data is the primary threat for cloud deployment in a multi-user/multi-tenant cloud environment. They further developed an algorithm to address the security issues of the cloud environment and proposed/applied a novel algorithm with the integration of Ciphertext Policy-Identity Attribute-based Encryption and the RSA algorithm for securing the cloud. Their research was able to establish a framework where both the owners and users are provided with the public and distinct secret keys that are generated by the Automated Certificate Authority. The proposed framework was implemented through Java. The performance of the proposed framework was analysed using standard metrices by comparing with the metrics output of Anand and Perumal, 2019 and Xue and Ren, 2019. However, the simulation of various data sizes revealed that the proposed framework is more expedient and effective when compared with EECDH and I-CP-ABE algorithms. The study also revealed that the proposed algorithm prevents man-in-the-middle attack. The authors adopted and applied RSA cryptosystem to the model, however, the RSA cryptosytem is not quantum safe.

Ref. [13] posited that the provision of data confidentiality and integrity of user's cloud data is subject to the provision of an effective security model that provides the mechanism that guarantees prevention of unauthorised access by third parties and a secured communication channel. The authors proposed a security framework that allows cloud users to handle the privacy and integrity of their data. The proposed model avails the user the opportunity to security, network usage, privacy, and data storage in the cloud without depending on the cloud provider. The model grants access to authorised and authenticated users to the cloud data, which has been proposed to be encrypted using a variant of AES algorithm. The proposed model was simulated using CloudSim with iFogSim as simulators on Eclipse integrated development environment. Results of the simulation revealed that the proposed model reduces energy consumption, network usage, and delay. Hence, the proposed framework enhances security, minimises resource utilisation, and reduces delay while utilising services of the cloud. The limitation of the

Table 2: Reviewed literatures

	Methodology	Remark
[15]	The McEliece cryptosystem was subjected to simulation in various extension degrees.	The private and public keys for the McEliece cryptosystem are very large matrices and consumes time in classical processing
[16]	Simulated NTRU, RSA, and AES to ascertain the performance of the cryptosystem	NTRU offers better performance when compared to other existing cryptosystems because the mechanism for encryption and decryption is simple
[17]	Used genetic algorithm to determine where data could be stored and adopted the capability list.	Consumes time in identifying the storage location of data
[18]	Used tornado codes and AES cryptosystem for cloud data security	AES has a simple key management system and encryption
[19]	Provided an overview of algorithms in lattice-based, super-singular elliptic curves, and code-based, and suggested adopting lattice-based algorithms	Computationally expensive and hence impractical to apply
[20]	Proposed adopting an RSA cryptosystem that generates primes in batches.	RSA cryptosystem will be broken by quantum computing
[21]	Applied binary method in Lattice multiplication to ECC cryptosystem	ECC increases the size of encrypted data and is not quantum safe
[22]	Proposed hierarchical role for access control	The proposed method cannot be applied to encrypted data
[23]	Applied RNS on NTRU	Any slight modification on the parameters of the framework slows down its computation
[24]	Proposed a four-level data cloud security, adopting IP filtering as the first level of security	IP cloning renders the model inefficient
[25]	The proposed framework adopted three cryptosystems namely the ECC for encryption, Dual-RSA algorithm for authentication, and MD5 for integrity check	ECC increases the size of the encrypted message. RSA is not quantum-safe
[26]	Proposed a Lattice-based Identity Based Encryption scheme with specific parameters	Keys are created and stored, which attracts hackers to attempt to hijack the system
[27]	Proposed the use of ECC to secure cloud applications	ECC possesses the tendency of producing encrypted data with large sizes
[28]	Proposed the adoption of HDFS in cloud computing environment	HDFS has very slow processing speed and does not support encryption at the data storage and network levels
[29]	Proposed using four prime numbers in the key generation of RSA	RSA is not quantum safe

study lies in the fact that the AES cryptosystem has key distribution challenges.

Ref. [14] suggested that cryptography is the most well-known technique for data security in a cloud environment. They further posited that cryptographic services in any cloud environment must accommodate authorisation, availability, confidentiality, integrity, and non-repudiation. They proposed the implementation of RSA, AES, and SHA256 in data security. The limitation of this mechanism is that it consumes a lot of time during execution. The long keys of RSA means that they incur high computational overhead and RSA cryptosystems are susceptible to quantum attacks. Furthermore, AES suffers from key exchange problem which is a limitation.

Ref. [30] researched on privacy and security challenges in a cloud environment in the context of a smart campus security with 100 respondents. The authors postulated that Blockchain technology overcomes the major challenge of cost in cloud computing. They further stated that cloud computing is less expensive and the blockchain technology is more expensive when applied with a chain of objects. More so, they opined that blockchain storage accounts are more secured, though security and privacy challenges depended on the type of business. Also, they stated that Blockchain technology can improve anonymity and security in a cloud environment and the major challenges include data integrity, data authentication, availability of data, location of data, data privacy, confidentiality, data storage, backup, and recovery. They concluded by positing that data integrity is affected because of inadequate encryption and audit control, authentication, and authorisation.

Ref. [10] stated that in spite of research works carried out in the area of cloud computing, challenges have persisted in the section of load balancing in cloud-based applications directed to the IaaS cloud service model. They postulated that IaaS model is technological driven that manages backend servers and virtual machine. Furthermore, they stated that cloud service providers should ensure situations where clients are being overloaded/underloaded to forestall machine failure or higher execution time suggesting task scheduling. The scholars proffered an LB algorithm directed towards optimising resources and enhancing load balancing considering the quality of service (QoS) task parameters, priority of virtual machines, and resource allocation. Results from their experiment revealed that the proposed LB algorithm had better execution time and makespan when juxtaposed with the Dynamic LBA algorithm.

The rest of the literatures reviewed are presented on tabular form (Table 2).

Upon review of relevant literature as regard to data security in cloud computing, it was discovered that most access control (username and password) data and user data were either secured using cryptosystems that will be rendered obsolete with the advent of quantum computing or not secured enough in the classical computing era. More so, most of the literature suggested the securing of both the user data and access control data using the same algorithm. Finally, literature revealed that authors adopted a hybrid cryptographic approach in the safeguarding of cloud data.

The following research gaps were identified based on the review;

Gap one: It can be deduced that most of the scholars/researchers neglect user authentication and how their data should be secured independently. Authentication and authorisation are key to any model proffered for cloud data security, since it decides who gets into a cloud system and what kind of data can be accessed. The adoption and application of authentication and authorisation on feature frameworks will go a long way in ensuring data security. Also, the independent encryption of access control data is also very important as most literature reviewed said little or nothing about this approach.

Gap two: It was revealed that quantum-safe cryptosystems were rarely adopted or applied in securing both access control data and cloud user data. Literature review also revealed that most of the frameworks proposed by authors are not/partially implemented or experimented on.

3 Materials and method

The article proposes a framework for an enhanced cloud data security. The proposed framework comprises a variant of the McEliece and NTRU algorithm. Subsequently, the algorithms are subjected to simulation with standard performance metrics alongside ECC, RSA, AES, and the existing NTRU and McEliece algorithms using MATLAB. The following subsections present the framework and the result of the simulation.

3.1 The proposed hybrid framework

In the proposed system, as depicted in Figure 3, a block diagram of the proposed framework is shown. McEliece cryptosystem is used to encrypt/decrypt user credentials, while the proposed NTRU is used to encrypt/decrypt user data.

The proposed framework for enhanced data security is presented in Figures 2 and 3 depicts the processes in the proposed framework. The cloud administrator creates users such that their user credentials are encrypted/decrypted using the proposed McEliece cryptosystem. Upon request to access the cloud, users provide their user credentials, which is subjected to authentication. It should be noted that the cloud administrator provides users with their credentials. The proposed model also includes the encryption and decryption of cloud data using a variant of the NTRU cryptosystem. The sections below explain the details of each segment of the model.

3.2 Proposed user verification

The framework proposes the authorisation and authentication of cloud users using onetime password (OTP) authentication and user credentials authentication. The figure below depicts the processes involved in the authentication and authorisation of cloud users (Figure 4).

For the proposed data storage/retrieval, an authenticated and authorised user accesses the data that have been stored in the cloud. Data is encrypted using a variant of the McEliece for user credentials and NTRU cryptosystems for user data and then stored/retrieved in the cloud.

3.3 Proposed McEliece cryptosystem

In a bid to increase the security of the McEliece cryptosystem, the key generation mechanism is strengthened, as shown below.

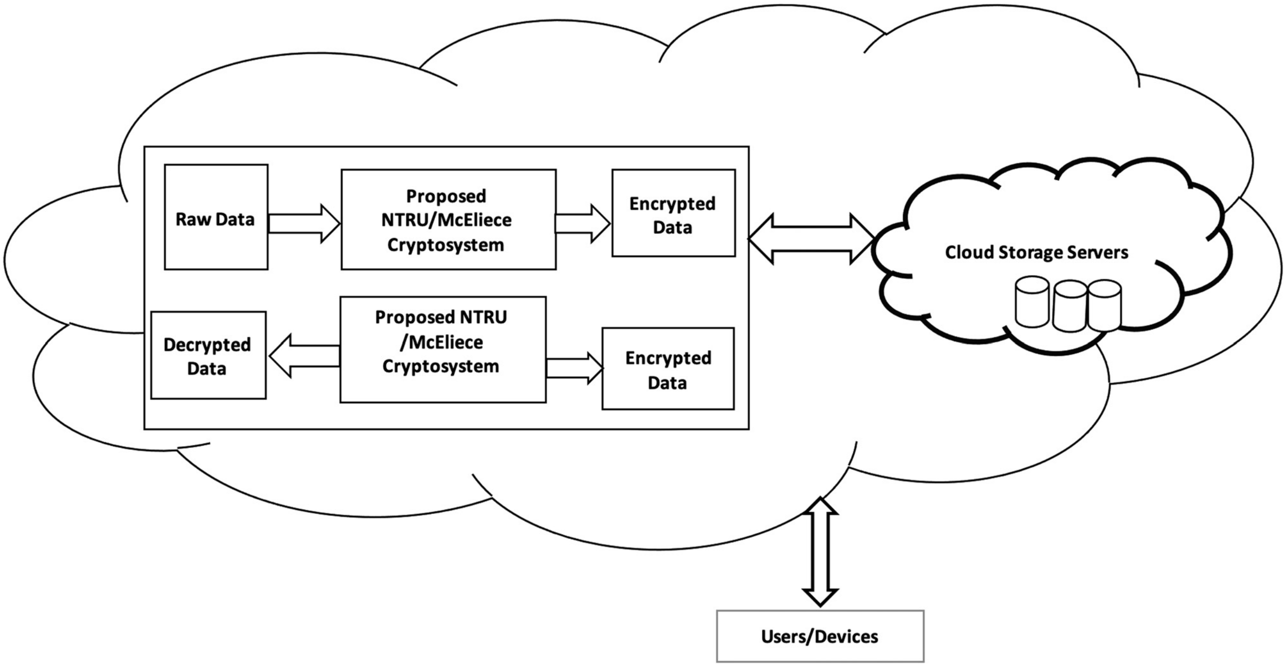

Figure 2: Optimised framework for data security in the cloud.

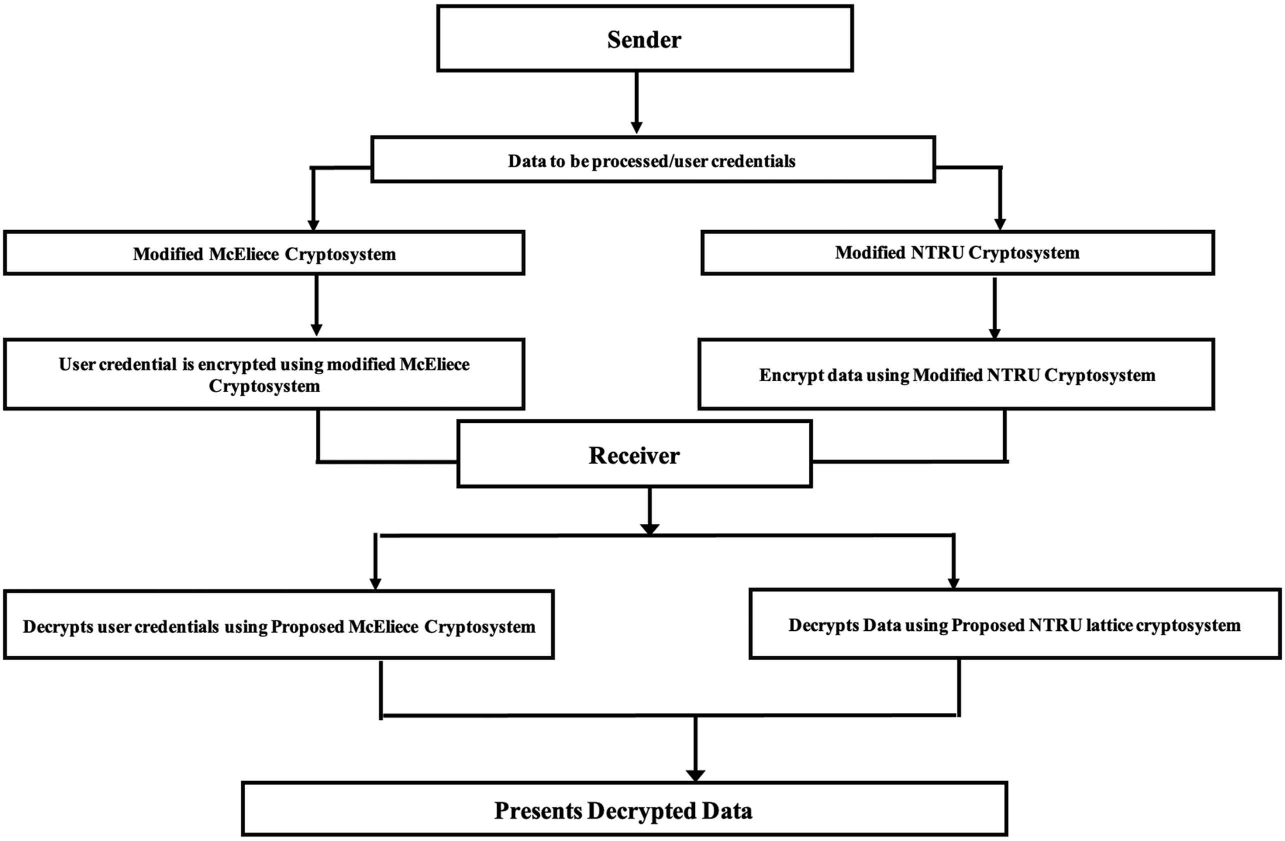
Sender
Data to be processed/user credentials
Modified McEliece Cryptosystem
Modified NTRU Cryptosystem
User credential is encrypted using modified McEliece Cryptosystem
Encrypt data using Modified NTRU Cryptosystem
Receiver
Decrypts user credentials using Proposed McEliece Cryptosystem
Decrypts Data using Proposed NTRU lattice cryptosystem
Presents Decrypted Data

Figure 3: A block diagram of the proposed framework.

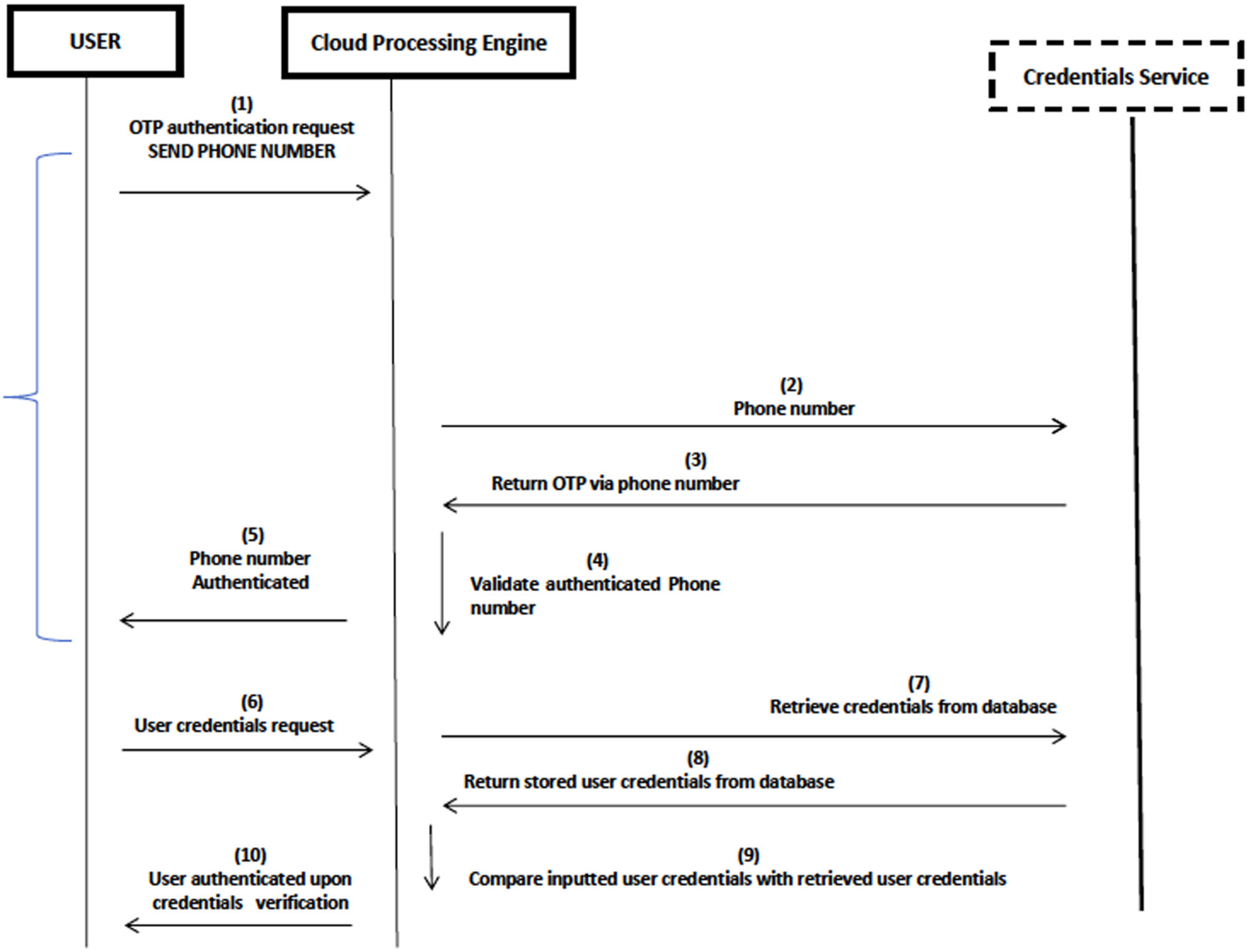
USER
Cloud Processing Engine
Credentials Service
(1)
OTP authentication request
SEND PHONE NUMBER
(2)
Phone number
(3)
Return OTP via phone number
(5)
Phone number
Authenticated
(4)
Validate authenticated Phone number
(6)
User credentials request
(7)
Retrieve credentials from database
(8)
Return stored user credentials from database
(10)
User authenticated upon credentials verification
(9)
Compare inputted user credentials with retrieved user credentials

Figure 4: Proposed user verification.

3.3.1 Definition of variables

G: Goppa codes, m: message to be encrypted

S: non-singular matrix, P: permutation matrix

P_k: public key, S_k: private key, t: weight of the distorting matrix, G_{key}: computed public key

The following are the proposed modifications to the existing McEliece algorithm:

McEliece consists of three algorithms: key generation algorithm, which produces a public and a private key, an encryption algorithm, and a decryption algorithm.

3.3.2 Key generation

Choose a binary Goppa code C, with parameters (n; k; and t). The generator matrix $G(n \times k)$ is obtained with a binary non-singular $(k \times k)$ matrix S and a permutation $(n \times n)$ matrix P, satisfying the following criteria; $S^T \neq S$, $S^{-1} \neq S$ and $P^T \neq P$.

The sender processes the encryption and decryption keys by processing

$$G_{key} = S^T \cdot G \cdot P^T, \tag{1}$$

where T represents transpose.

The sender also sets the public key as:

$$P_k = (G_{key,}\ t) \text{ and private key as : } S_k = (S^T, G, P^T).$$

3.3.3 Encryption

The sender's message (m) is encoded in binary form with a length, k. The following is processed by the sender

The sender processes:

$$C^I = m \times G_{key}. \tag{2}$$

A random n-bit vector z, containing exactly t ones (a vector of length n and weight t), is generated by the sender and then the ciphertext is computed as:

$$C = C^{\mathrm{I}} + z. \tag{3}$$

3.3.4 Decryption

The receiver upon receiving the ciphertext computes the inverse of P^{T} i.e., $(P^{\mathrm{T}})^{-1}$ and computes:

$$d = C(P^{\mathrm{T}})^{-1}. \tag{4}$$

The receiver finally uses the decoding algorithm for the code G to decode d to X and computes:

$$m = X(S^{\mathrm{T}})^{-1}, \tag{5}$$

which is equivalent to the original message.

Figure 5 below depicts the processes involved in the proposed McEliece algorithm.

Below is the algorithm for key generation, encryption, and decryption of the proposed McEliece algorithm.

Algorithm 1: Proposed McEliece-key generation

```
Input: parameters for key generation ($S$, $P$, $G$)
Output: keys ($G_{\mathrm{key}}$, $P_{\mathrm{k}}$)
Begin
  i. Calculate the transpose of $S$ and $P$
  ii. Compute $G_{\mathrm{key}} = S^{\mathrm{T}} \cdot G \cdot P^{\mathrm{T}}$
  iii. $P_{\mathrm{k}} = (G_{\mathrm{key}}, t)$
  iv. Return ($G_{\mathrm{key}}$, $P_{\mathrm{k}}$)
End
```

In the above algorithm, the sender sets the public key as $P_{\mathrm{k}} = (G_{\mathrm{key}}, t)$ and the private key as $S_{\mathrm{k}} = S^{\mathrm{T}} \cdot G \cdot P^{\mathrm{T}}$. Below is the encryption algorithm for proposed McEliece cryptosystem.

Algorithm 2: Proposed McEliece-encryption

```
Input: Parameters for encryption ($m$, $G_{\mathrm{key}}$)
Output: Cipher text (C)
Begin
```

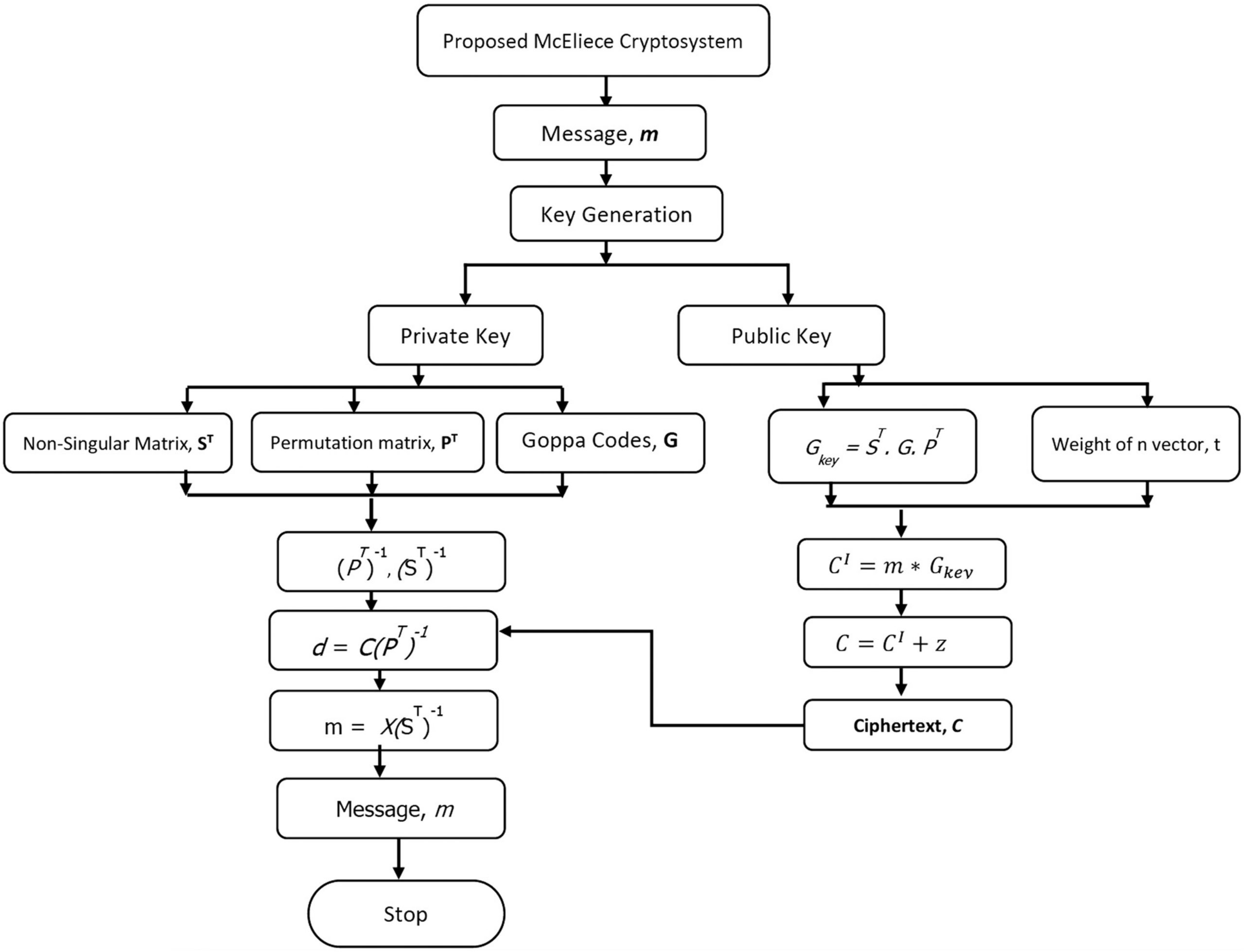

Figure 5: Proposed McEliece cryptosystem.

```
    i. Compute $C^{\mathrm{I}} = m \times G_{\mathrm{key}}$
    ii. A random $n$-bit vector $z$, containing exactly $t$ ones
(a vector of length $n$ and weight $t$, is generated)
    iii. $C = C^{\mathrm{I}} + z$
    iv. Return (C)
End
```

Below is the proposed decryption algorithm for McEliece cryptosystem.

Algorithm 3: Proposed McEliece-decryption

```
Input: Parameters for encryption ($S$, $P$, $G$)
Output: Plain text ($m$)
Begin
    i. Computes the inverse of $P^{\mathrm{T}}$ i.e., $(P^{\mathrm{T}})^{-1}$
    ii. Use the decoding algorithm for the code $G$ to
decode $d$ to $X$
    iii. $m = X(\mathrm{S}^{\mathrm{T}})^{-1}$
    iv. Return ($m$)
End
```

3.4 Proposed NTRU cryptosystem

Data is encrypted/decrypted using the modified NTRU cryptosystem, the proposed NTRU cryptosystem will be driven by lattice arithmetic. Lattice multiplication is the method for multiplying bigger numbers or for carrying out complex multiplication. It is algorithmically identical to the traditional long multiplication method, but breaks the process into smaller steps. The following steps are required for the proposed NTRU cryptosystem.

3.4.1 Definition of variables

N – the polynomials in the ring R with degree $N - 1$.

p and q – are small and large modulus respectively, which are used for the reduction in coefficients in the encryption/decryption of data.

f and g – polynomials used to process the public key h, r – a random blinding polynomial used to distort data, m – is the message to be encrypted/decrypted represented in polynomial form.

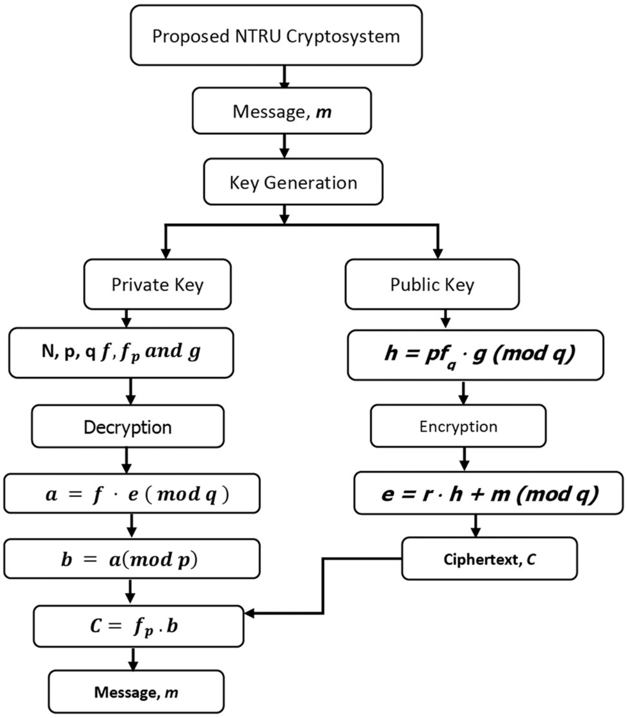

Figure 6: Proposed NTRU cryptosystem.

Table 3: Total average execution time taken for the proposed McEliece

File size (kB)	ECC (s)	AES (s)	RSA (s)	Existing McEliece (s)	Proposed McEliece (s)
20	0.0102	0.5501	0.1600	17.5009	17.6744
77	0.0302	0.4901	1.1400	64.4509	66.0208
153	0.0202	0.5001	1.2600	141.9506	133.5506
283	0.0402	0.5001	1.4900	250.9907	247.8208
305	0.0402	0.5301	2.3500	268.8508	266.3308
Average time	0.02820	0.5141	1.2800	148.7488	146.2795
Throughput	20856.1473	1580.7977	356.5955	3.1170	3.1465

Upon access to the cloud, users are given the opportunity to either edit already uploaded data or upload fresh data, these data are secured using the proposed NTRU cryptosystem. NTRU algorithms consists of three algorithms: key generation algorithm, which produces a public and a private key, an encryption algorithm, and a decryption algorithm. The following processes are driven by lattice arithmetic.

3.4.2 Key generation

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

$$h = pf_q{\cdot}g(\text{mod } q). \tag{6}$$

3.4.3 Encryption

To encrypt a message, the following is processed:

$$e = r{\cdot}h + m(\text{mod } q). \tag{7}$$

3.4.4 Decryption

The following is computed to decrypt the message

$$a = f{\cdot}e(\text{mod } q), \tag{8}$$

$$b = a(\text{mod } p), \tag{9}$$

$$C = f_p \cdot b, \tag{10}$$

which is equivalent to the original message.

The figure below shows the processes involved in the encryption and decryption of NTRU cryptosystem driven by lattice arithmetic (Figure 6).

Below are the algorithms for processes of the proposed NTRU cryptosystem.

Algorithm 4: Proposed NTRU-key generation

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

Algorithm 5: Proposed NTRU-encryption

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

Algorithm 6: Proposed NTRU-decryption

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

4 Results and discussion

The proposed algorithms were subjected to limitation against ECC, AES, RSA, existing McEliece, proposed McEliece, NTRU, and proposed NTRU algorithms.

Table 4: Total execution time taken for the proposed NTRU

File size (kB)	ECC (s)	AES (s)	RSA (s)	Existing NTRU (s)	Proposed NTRU (s)
20	0.0133	0.5539	0.1573	811.0032	2024.7362
77	0.0136	0.4875	1.1636	3243.864	8176.1374
153	0.0219	0.5045	1.2572	6560.97	19572.5671
283	0.0393	0.4987	1.4974	12224.2308	44397.6335
305	0.0352	0.5272	2.3517	37438.7863	97741.92
Average execution time	0.02466	0.5144	1.2854	12055.7707	34382.5988
Throughput	33976.10	1629.19	651.92	0.07	0.02

Table 3 above shows the execution time of the proposed McEliece algorithm as well as the average execution time and throughput of the simulated algorithms.

Table 3 above shows that ECC has the best throughput. Comparing the existing and proposed McEliece algorithms, it is revealed that the proposed McEliece algorithm has a higher throughput and a lower average execution. Table 4 below shows the execution time of the proposed NTRU algorithm as well as the average execution time.

Table 4 above depicts that ECC has the best throughput. Juxtaposing the existing and proposed NTRU algorithms, it is revealed that the existing NTRU algorithm has a higher throughput and a lower average execution.

In comparing, the existing and proposed McEliece algorithm, the above proved that the introduction of the transpose function of variables *S* and *P* in the proposed McEliece algorithm enhanced the total execution time of the algorithm. However, the proposed NTRU algorithm, which is Lattice multiplication driven, proved to be more time consuming and with lower throughput. It can thus be deduced that the proposed McEliece is effective and efficient while that of the proposed NTRU is less effective and efficient.

5 Conclusion

This article proposes a framework that adopts quantum-safe algorithms to safeguard cloud data. Innovatively, a variant of McEliece cryptosystem was used to safeguard user credentials, while a variant of NTRU cryptosystem was used to safeguard cloud data. The McEliece and NTRU cryptosystems were proposed to provide an efficient data security in the cloud environment amidst the emergence of quantum computing. It is expected that the proposed model will decrease man-in-the-middle attacks and improves data security.

6 Future work

This article highlights post-quantum cryptographic systems and recommends the application of the proposed algorithms on a quantum computer platform as the authors of this article carried out its simulation on a classical computer.

Conflict of interest: Authors state no conflict of interest.

Data availability statement: The authors declare that data supporting the findings of this study are available within the article.

References

[1] M. N. Daodu, A. Gabriel, B. K. Alese, and A. O. Adetunmbi, “A data encryption standard (DES) based web services security architecture,” *Ann Comput Sci Series, Tibiscus Univ*, vol. 14, no. 2. pp. 53–8, 2016.

[2] B. K. Alese, Deign of public key cryptosystem using elliptic curve, *Thesis*, Akure, Ondo State, Nigeria, The Federal University of Technology, 2004.

[3] A. F. Thompson, O. E. Oyinloye, M. T. David, and B. K. Alese, “A secured system of Internet Enabled Host Devices,” *Netw Commun Technol*, vol. 5, no. 1. pp. 26–36, 2020.

[4] A. J. Gabriel, B. K. Alese, A. O. Adetumbi, and O. S. Adewale, “Post-quantum cryptography based security framework for Cloud Computing,” *J Internet Technol Secured Trans*, vol. 4, no. 1. pp. 351–7, 2015.

[5] A. M. Kuo. *Opportunities and Challenges of Cloud Computing to Improve Health Care Services*. 2011. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3222190/. Accessed 02/03/2018.

[6] S. S. S. Shehata, *Post Quantum Cryptography with Random Split of St-Gen Codes*, Norwegian University of Science and Technology, Department of Information Security and Communication Technology, 2017. Available at: https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/2450584/16929_FULLTEXT.pdf?sequence=1.

[7] J. Buchmann, and J. Ding, “Post-quantum cryptography”, *Second Int Workshop*, PQCrypto, 2008, pp. 17–9.

[8] L. Chen, S. Jordan, Y. -K. Liu, D. Moody, R. Peralta, R. Perlner et al., Report on post-quantum cryptography, *National Institute of Standards and Technology Internal Report 8105*, 2016. Available at: https://dl.acm.org/doi/proceedings/10.5555/1473109.

[9] D. Micciancio and O. Regev *Lattice-based cryptography*. 2008. Available at http://cims.nyu.edu. Accessed 28/06/2020.

[10] D. A. Shafiq, N. Z. Jhanjhi, A. Abdullah, and M. A. Alzain, “A load balancing algorithm for the data centres to optimise cloud computing applications,” *IEEE Access*, vol. 9, pp. 41731–44, 2021, doi: 10.1109/ACCESS.2021.3065308.

[11] G. Summers, Data and databases, *Developing Databases with Access*, H. Koehne, editor, Nelson Australia Pty Limited, 2004, pp. 4–5. Available at: https://catalogue.nla.gov.au/Record/4610312.

[12] S. Chandel, G. Yang, and S. Chakravarty, “RSA-CP-IDABE: a secure framework for multi-user and multi-owner cloud environment,” *Information*, vol. 11, p. 382, 2020.

[13] I. J. Awan, M. Shiraz, M. U. Hashmi, Q. Shaheen, R. Akhtar, and A. Ditta, “Secure framework enhancing AES algorithm in cloud computing,” *Hindawi, Security Commun Netw*, vol. 2020, pp. 1–16, 2020.

[14] M. M. Abdelnapi, F. A. Omara, and N. F. Omra, “A hybrid hashing security algorithm for data storage on cloud computing,” *Int J Comput Sci Inf Security* (*IJCSIS*), vol. 14, no. 4, pp. 175–181, 2016.

[15] 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.

[16] 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.

[17] S. Mall, and K. Saroj, “A new security framework for cloud data”, *8th International Conference on Advances in Computing and Communication* (*ICACC*), 2018.

[18] R. Wang, Research on data security technology based on cloud storage, *13th Global Congress on Manufacturing and Management, GCMM 2016*, 2016.

[19] M. Kindberg, “A usability study of post-quantum algorithms”, *Master’s thesis*, Lund, Sweden, Department of Electrical and Information Technology Lund University, 2017, Retrieved from https://pdfs.semanticscholar.org/8ed3/7b0e436e96384bfb14f02ea21c9a9f84ee65.pdf. Accessed 21/09/2020.

[20] D. J. Bernstein, N. Heninger, P. Lou, and L. Valenta, *Post-quantum RSA*, 2017, Available at https://cr.yp.to/papers/pqrsa-20170419.pdf. Accessed 04/06/2018.

[21] S. Pavithra and S. Baskar, “Lattice based multiplier for WSN applications for ECC,” *Int J Trend Res Dev*, vol. 2, no. 6, pp. 21–27, 2015.

[22] P. Zhang, J. Xu, H. Muazu, and W. Mao, “Access control research on data security in cloud computing”, *2015 IEEE 16th International Conference on Communication Technology* (*ICCT*), Hangzhou, China, 2015, pp.874–44.

[23] A. Zalekian, M. Esmaeildoust, and A. Kaabi, “Efficient implementation of NTRU cryptography using residue number system,” *Int J Comput Appl* (*0975 – 8887*), vol. 124, no. 7. pp. 33–7, 2015.

[24] A. Siam, H. El-khobby, S. Abd Elkader, and M. AbdelNaby, “Enhanced data security model for cloud computing platform,” *Int J Sci Res Science, Eng Technol*, vol. 1, no. 4, pp. 450–460, 2015.

[25] D. V. Kumar, “A hybrid security protocol using python,” *Int J Comput Sci Inf Technol Res*, vol. 2, no. 4, pp. 9–16, 2014.

[26] L. Ducas, V. Lyubashevsky, and T. Prest, *Identity-based encryption NTRU lattices*, 2014, https://eprint.iacr.org/2014/794. Accessed 28/07/2020.

[27] O. D. Alowolodu, B. K. Alese, A. O. Adetunmbi, O. S. Adewale, and O. S. Ogundele, “Elliptic curve cryptography for securing cloud computing applications,” *Int J Comput Appl* (*0975 – 8887*), vol. 66, no. 23, pp. 11–17, 2013.

[28] P. Dhawan, “Data security model for cloud computing,” *Int J Res Sci Technol* (*IJRST*) *2013*, vol. No. 2, no. V, pp. 264–271, 2013.

[29] U. P. B. Ivy, P. Mandiwa, and M. Kumar, “A modified RSA cryptosystem based on ‘n’ prime numbers,” *Int J Eng Comput Sci*, vol. 1, no. 2. pp. 63–6, 2012, ISSN:2319-7242.

[30] S. H. Gill, M. A. Razzaq, M. Ahmad, F. M. Almansour, I. Haq, N. Z. Jhanjhi, et al., “Security and privacy aspects of cloud computing: a smart campus case study,” *Intell Autom Soft Comput*, vol. 31, no. 1, pp. 117–128, 2022, doi: 10.32604/iasc.2022.016597.

[31] H. Kharche and D. S. Chouhan, “Building trust in cloud using public key infrastructure,” (*IJACSA*) *Int J Adv Comput Sci Appl*, vol. 3, no. 3. p. 2012, 2012.