*International Journal of Engineering & Technology*, 7 (2.33) (2018) 813-817

International Journal of Engineering & Technology

*Website: www.sciencepubco.com/index.php/IJET*

***Research paper***

# A P-NTRU and secret key sharing techniques of access control scheme for big data storage in cloud

**S. V. Divya [1] *, Bala Monika [2], Venkadesh [3]**

*[1] Department of Information Technology, Noorul Islam Centre for Higher Education, Kumaracoil-629180, Thuckalay, TN*
*[2] P.G Student, Noorul Islam Centre for Higher Education, Kumaracoil-629180, Thuckalay, TN*
*[3] Department of CSE, Noorul Islam Centre for Higher Education,Kumaracoil-629180,Thuckalay,TN*
**Corresponding author E-mail: divyasadasivam@gmail.com*

## Abstract

Outsourcing the ciphertexts in the cloud storage server becomes an effective way for storing the big data. However, verifying the access policy of the user and updating it regularly in the cloud server makes it really challenging. Various approaches have been proposed but they do not offer a complete solution. Hence, an efficient and secure proposed NTRU (P-NTRU) is proposed. Moreover, the P-NTRU system efficiently updates the stored cipher-text in the server when the owner's access policy is changed. The proposed P-NTRU system enables the owner to verify the legitimacy of the user. With the introduction of the certificate authority, the burden of the data owner is highly reduced. Analysis proves that the proposed P-NTRU system is efficient and secure.

***Keywords***: *Access Policy; Certificate Authority; Legitimacy; NTRU.*

## 1. Introduction

Big data [1] in cloud requires a massive amount of storage, velocity and variety of information and needs a contemporary form of information processing. Due to its massive capacity and large volume, managing data in big data is of course difficult. Hence the data should be handed over to a cloud server that has the adequate potential to handle them.

The traditional approaches are normally based on attribute-based encryption [2-5] or secret sharing techniques. This approach enable flexibility for the cloud data owners and also it includes a list of predefined users who are has the right to access those data. However these approaches suffer from updating the access policies and handling the ciphertext.

Another important concern is sharing a secret key between the two entities by means of secret sharing techniques [6-7]. Here, the secret key is reconstructed by the cooperative users who generally employ a cryptographic techniques such as an RSA algorithm. This causes the computational overhead to be higher. Further, challenges lies in updating the access policies as per the user needs.

Since the data owners do not keep a backup of his data locally after outsourcing, it is tedious to manage the data stored in the cloud server. Besides, as enterprises are shifting their data rapidly into the cloud environment, it is difficult to manage and update the access policies and also to deal when the users' join or leave the group.

Verifying the legitimacy of the user who accesses those outsourced data in the cloud server is another major issue. Various approaches have been proposed to verify the legitimacy of the user but majority of those methods does not have enough mechanism to validate the user. Also, they rely on RSA algorithms which increase the computational complexity because of the involvement of multiple operations.

Moreover, the asymmetric cryptosystems were merely to suffer from quantum computing attack resistance in the near future. Therefore an NTRU cryptosystem [8-10] was proposed to overcome the quantum computing attack but suffers from decryption problems.

## 2. Background works

As the amount of data is growing rapidly in today's environment, managing those data is also tedious. It is a challenging issue that how users store, manage, and analyze those huge data in a timely and cost-effective way.

As businesses are moving their confidential data into cloud, how to manage those outsourced data and how to store those data effectively become the major issues. Before outsourcing the data, owners generally employ cryptographic techniques to encrypt the data. Moreover, the data owner also ensures the correctness of the cloud data periodically.

An effective mechanism during outsourcing should be provided for a smooth cooperation between the data owners and the server. Access control mechanisms were generally employed to safeguard the stored data in the cloud. An attribute- based encryption [11] scheme which defines the access policies based on the attributes of the user was proposed. Since the data owners do not have a backup copy locally after outsourcing, it is difficult to update the access policies and it also incurs high computation overhead and energy consumption.

Inorder to protect the massive amount of data in cloud storage, secret sharing mechanisms [12] were generally employed. Two schemes based on RSA and linear recursion [13] was proposed. To validate the users' access rights, RSA is used whereas the later scheme is used for secret key generation.

In [14], a multi-secret sharing scheme based on cellular automata was proposed. The scheme is used to construct the secret key share based on RSA cryptosystem with a linear computational complexity. Since there is a need for multiple users to verify each other users mutually, there involves multiple RSA operations and hence incurs high computational overhead. Also, the traditional schemes were not able to satisfy the requirements of quantum computing. Hence NTRU cryptosystem [15] to overcome the quantum computing attacks was proposed. Moreover, a secret key share mechanism was also employed for validating the users' access legitimacy but it suffers from decryption problems. In [16], security in data forwarding in cloud is achieved by means of homomorphic encryption and an online alert scheme is introduced when unauthorized user tries to access the data.

Therefore an efficient and verifiable NTRU cryptosystem which checks the users' access legitimacy is necessary. It should also overcome the decryption problems in the original NTRU cryptosystem and highly resistant to quantum computing attacks.

## 3. Contribution of the paper

1) An efficient and secure proposed P-NTRU system is proposed.
2) The P-NTRU system is used to check the legitimacy of the user.
3) A P-NTRU system is used to overcome the failure problems and collusion attack which prevails in the existing scheme.
4) The P-NTRU cryptosystem allows updation of the cloud data as per the access rights of the user.

## 4. Preliminaries and notations

The preliminaries and the notations used in the proposed NTRU cryptosystem are defined below:

### 4.1. System setup

Assume there are n messages M= {$M_1$, $M_2$, .$M_n$} and there exists t users and B represents the set of users such that B= {$U_1$, U2, Ut}.

### 4.2. NTRU cryptosystem

The NTRU [9-10] cryptosystem is proposed to overcome the quantum computing attacks. To overcome the drawbacks which exist in the traditional cryptographic algorithms, NTRU system based on shortest vector problem was introduced .NTRU is much faster than the RSA algorithm.

### 4.3. Threshold secret sharing technique

The secret sharing [12, 15] is an important technique to transform the secret key between the owners and the users. The user gets the key if and only if it cooperates mutually with all other users. For example if the owner needs to generate the secret key for the user,

1) Generate a polynomial function.
2) Construct the secret key share

### 4.4. Notations

The notations and the semantic meaning used in the proposed NTRU cryptosystem are depicted in Table 1.

**Table 1:** Notations

| Notations | Meaning |
|---|---|
| D | Data Owner |
| U | User |
| S | Message Set |
| R | Ring |
| $L_a$ , $L_b$ | Set of polynomials in R |
| f | Private key |
| k | Public key |
| $X_{ij}$ | Exchange certificate |
| Φ | Random parameter |

## 5. Modelling and architecture

The proposed cloud storage system is appropriate for both private and public clouds. The proposed cloud storage system consists of three entities: the cloud/cloud server, data owners and the users and it is shown in Figure 1.

- Cloud/Cloud server: The cloud server allocates space for storing the owners' data especially the cipher text. It is also responsible for updating the stored cipher text when the owners change their access policy.

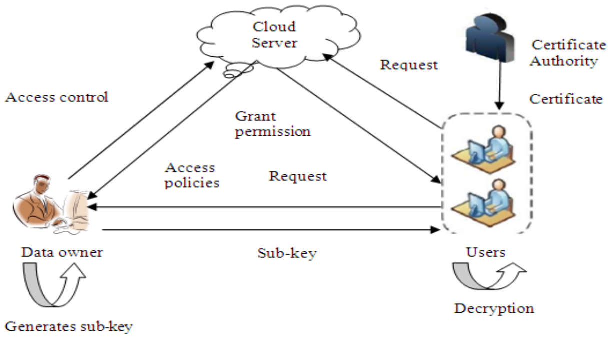

**Fig. 1:** Architectural Model.

- Data Owner: It is the responsibility of the data owner to assign the access policy for his data before outsourcing it. He also validates the stored cloud data when the access policy of the user changes.
- User: The user gets a sub-key from the owner to access his eligible data. Suppose of there are n users, each user is verified by (n-1) users. Also to decrypt a message, the information is again verified by means of secret key sharing technique.

### 5.1. Phases in NTRU cryptosystem

The proposed NTRU cryptosystem consists of 5 phases:

#### 5.1.1. Key generation

The public and the private keys are generated here. Choose two polynomials p and f such that pЄ$L_p$ and fЄ$L_f$. Let f be the private key and has two inverse modulo $f_q$ and $f_p$ which is computed as,

$$Fq*f=1 \text{ (mod q) and } f_p*f=1 \text{ (mod p)} \quad (1)$$

If the private key is properly selected, then it is quite easy to compute $f_q$ and $f_p$. In the proposed system, Elliptic curve cryptography is used for key generation purpose. Select a random number d such that it lies within the ECC range [1 to n-1] and P is the point on the curve. Now the public key k is computed as:

$$k=d*f_q*g \text{ (mod q)}*P \quad (2)$$

Sends his public parameters (p,q, k).

#### 5.1.2. Sub-key generation

For each user, the data owner computes t different integers such that $n_0$, $n_1$,....$n_{t-1}$ and uses them as coefficients to construct the degree polynomial n(x).

$$N\ (x) = n_0+\sum_{j=1}^{t-1} n_j x^j \quad (3)$$

Where $n_j$ЄR, RЄZ|X| / $(X^N$ -1)

### 5.1.3. Cipher text generation

Each user selects a random integer r where r=one...i, and encrypt it using the data owner’s public key k and generates the cipher text c as:

$$C=p\Phi*k+r_i\,(modq) \tag{4}$$

After generating the cipher text, the user sends $\{id_i, c, H(r_i)\}$ to the data owner; where c is the generated cipher text and the $H(r_i)$ is the hash of the random number.
When the data owner receives the cipher text c, it is decrypted using the private key f and get the plain text and checks whether:

$$H(r_i^{'}) = H(RI) \tag{5}$$

If the condition is satisfied, the data owner generates a sub key $y_i$ for the user $U_i$.

$$Yi= b\,(RI) = \sum_{j=0}^{t-1} n_j (RI)^{j} \tag{6}$$

And securely broadcasts $\{id_i , x_i, H(id_i \,||\, r_i)$ to all the users.

## 5.2. Certificate construction

The certificate authority (CA) computes a certificate to validate the user for accessing the data. A copy of the certificate is also sent to the data owner for validating the user .The CA randomly selects a parameter $\Phi \in L_{\Phi}$ and $e= \{e_1, e_2, .e_j, eM\}$ where $e_j \in R$ and computes a certificate $(e_j,d_j)$ for all the messages.

$$D_j=p\Phi*f+e_i\,(modq) \tag{7}$$

Where

j= 1 to M

## 5.3. Encryption

The data owner computes a set of ciphertext T = {t1, t2, ti...tj} for all the user messages S= {s1, s2, si,sj} before outsourcing it to the cloud server. Each owner messages has a point M on the ECC curve.

$$Ti = Si \oplus H\,(n0*dj) \tag{8}$$

After computing ti, the data owner performs Elliptic curve cryptography for encrypting the plain text into cipher text. For each message to be transmitted, two cipher texts were generated. Let it be CT1 and CT2.

$$CT1=d*P \tag{9}$$

$$CT2=M+ (d*P) \tag{10}$$

Where CT1 and CT2 were stored in Cipher text set CT. i.e CT=CT1+CT2

## 5.4. Exchange certificate

To retrieve a message from the message set, initially the user sends a request to the data owner to get the message certificate. Upon receiving, the data owner encrypts the certificate using the user’s secret random number by using AES algorithm and computes the cipher text CT set. All the generated cipher texts were stored in the cipher text set.

$$CTdj= AESri\,(dj) \tag{11}$$

After computing the cipher text from the data owner, the user decrypts it to obtain $d_j$. The exchange certificate $X_{ij}$ is computed as,

$$X_{ij}= y_i * d_j \tag{12}$$

## 5.5. Certificate validation

When a user receives in t receives the exchange certificate $X_{ij}$, it then uses $d_j$ to verify such that:

$$X_{ij}=y_i*d_j \tag{13}$$

And then compute $H^{'}= H(id_i \,||\, r_i)$ and checks whether $H^{'}=H$ since all the users contain those information.

## 5.6. Message reconstruction

The message is reconstructed here when a user obtains the authorization information from all other (t-1) users. The message S is reconstructed from (8) as,

$$S_i= t_i \oplus H\,(n_0*d_j)$$

$$S_i=t_i \oplus CT_{dj} \oplus H\left(\left( \sum_{u_{i \in B}} x_i * \prod_{u_{\partial \in B, u\partial \neq B}} \frac{r_\partial}{r_{\partial - r_i}}\right)*d_j\right) \tag{14}$$

Where $u_\partial$ is any user and B the set of users.

## 5.7. Policy updation

The stored cloud data in the server is updated as the access rights changes. The newly generated access policy is computed by:

$$n^{'}(x) = n_0^{'} + \sum_{j=1}^{t'-1} n_j^{\prime} x^{j} \tag{15}$$

Where $n_0^{'}$ is the new polynomial of $n^{'}(x)$ . The sub-key $y_i^{'}$ and the message certificate $d_j^{'}$ are generated by the data owner and the certificate authority respectively. For updating the access policy, the data owner computes an intermediate value $l_{iv}^{'}$ as,

$$l_{iv}^{'}=H(n_0*d_j) \oplus H(n_0^{'}*d_j^{'}) \tag{16}$$

And sends it to the cloud server and request to update the data stored in it. The cloud server computes

$$t_i^{'}=t_i \oplus l_{iv}^{'} \tag{17}$$

And sends it to the data owner. Upon receiving the value, it is the responsibility of the data owner to validate the server for access policy updation. The server validation is performed by,

$$H_1(S_i) = H_1(t_i^{'} \oplus H\,(n_0^{'}*d_j^{'})) \tag{18}$$

## 5.8. Decryption

Since the entire owner cipher text messages CT are mapped with the ECC curve point M, decryption is performed by,

$$M= (S_i\text{-}CT2)–CT1 \tag{19}$$

## 5.9. Steps in NTRU cryptosystem

Step1: The public and the private keys are generated using the Key Generation algorithm using ECC using (1) and (2).
Step2: A degree polynomial is generated using (3).
Step 3: The cipher text is generated using (4).
Step4: The user then sends his id, the generated cipher text and the random number to the owner.
Step5: The data owner checks the condition using (5)
Step6: If the condition is true, the data owner generates a sub key using (6).
Step7: For validating the user, the data owner computes a certificate using (7).

Step 8: The data owner then computes a set of cipher text for all the user messages using the (8), (9) and (10).
Step 9: The exchange certificate is generated using (11) and (12).
Step10: The generated certificate is validated using (13).
Step11: The original message is reconstructed using (14).
Step12: The encrypted data is updated in the cloud server according to (15)-(18).
Step13: The message is decrypted using (19).

## 6. Results and discussions

To analyze the performance of the proposed P-NTRU system, various experiments were conducted based on various metrics such as key generation time, data security and time efficiency. The implementation is done using Java language in Intel dual core processor which has 4 GB RAM memory and 2 GHZ speed.

### 6.1. Performance analysis

The performance of the proposed NTRU system is compared with the existing NTRU system with the following metrics:

#### 6.1.1. Key generation time

In the existing NTRU system, the private key, the public key and the sub key is generated which is used to compute the exchange certificate. But in the proposed NTRU system, the sub key is used to compute the cipher text as well as certificate. So the P-NTRU system requires more time as the data size increases.

**Table 2:** Data Size vs Time

| Data Size (MB) | Time (ms)<br>NTRU System | P-NTRU System |
|---|---|---|
| 1 | 5 | 8 |
| 2 | 9 | 12 |
| 3 | 10 | 15 |
| 4 | 12 | 18 |
| 5 | 20 | 25 |

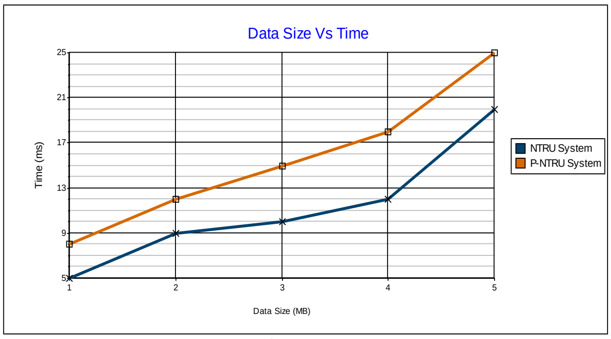

**Fig. 2:** Graph for Data Sizes vs Time.

Similarly, in the P-NTRU system, for each and every plain text, two cipher text were generated and it is stored in the cipher text sets. Hence the key generation time for the P-NTRU system is high which is shown in Table 2 and Figure 2.

#### 6.1.2. Computational cost

The communication cost of the existing NTRU system is compared with that of P-NTRU system for different file size.

**Table 3:** File Size vs Computational cost

| File Size (KB) | Computational cost (bits/sec) | |
|---|---|---|
| | NTRU System | P-NTRU System |
| 10 | 1.21 | 1.18 |
| 15 | 2.20 | 2.15 |
| 20 | 2.26 | 2.22 |
| 25 | 2.34 | 2.31 |
| 30 | 2.36 | 2.32 |

The communication cost of the proposed P-NTRU system is compared with that of existing NTRU system by considering the

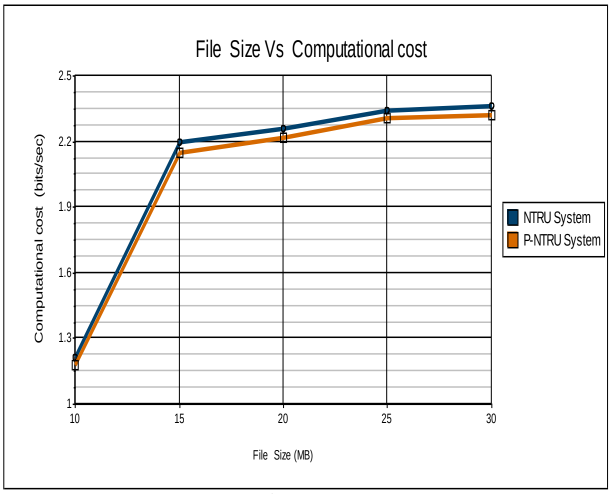

**Fig. 2:** Graph for Data Sizes vs Time.

## 7. Conclusion

In this paper, an efficient and improved P-NTRU cryptosystem is proposed to overcome the failures in the original NTRU cryptosystem.

The improved P-NTRU system is capable of efficiently storing the big data in the cloud storage server and performs all the basic functionalities of the cloud storage. It also enables the server to update the stored data when the access policy of the owner varies. Moreover, the computational burden of the data owner is highly reduced by introducing the certificate authority also the proposed system provides better security with the intrusion of ECC cryptography with less key size. As a future enhancement, the scheme can be applied to multi-cloud storage environments and analyze the security problems that exists when the data owner outsources the data into multi-cloud servers.

## References

[1] M.A.Beyer and D.Laney, “The importance of bug data: a definition”, Stanford, CT: Gartner, 2012.
[2] A.Sahai and B.Waters, “Fuzzy identity based encryption”, Advances in Cryptology-EUROCRYPT 2005, pp. 457-473, 2005.
[3] V.Goyal, O.Pandey, A.Sahai and B.Waters, “Attribute-based encryption for fine-grained access control of encrypted data” , in Proceedings of the 13th ACM conference on Computer and Communications Security, ACM 2006,pp.89-98.
[4] C.Hu,X.Cheng ,Z.Tian,J.Yu,K.Akkaya and L.Sun, “An attribute-based signcryption scheme to secure attribute-defined multicast communications”, in Secure Communications 2015,Springer ,pp.418-435,2015.
[5] B.Waters, “Ciphertext –policy attribute based encryption: An expressive, efficient and provably secure realization”, Public key cryptography-PKC 2011, pp.53-70, 2011.
[6] M.H.Dehkordi and S.Mashhadi, “New efficient and practical verifiable multi-secret sharing schemes”, Information Sciences, Vol.178, no.9, pp.2262-2274, 2008.
[7] J.Zhao,J.Zhang and R.Zhao, “A practical verifiable multi-secret sharing scheme”, Computer Standards & Interfaces,vol.29,no.1,pp.138-141,2007.
[8] O.Regev, “New lattice-based cryptographic constructions”, Journal of the ACM (JACM), vol.51, no.6, pp.899-942, 2004.
[9] J.Hoffstein, J.Pipher , and J.Silverman, “ NTRU :A ring-based public key cryptosystem”, in Algorithmic number theory:third international symposium, ANTS-III, Portland,Oregon, USA,June 21-25,1998:Proceedings,vol.1423, Springer Verlag,1998,pp.267-288
[10] N.Cryptosystems, “The NTRU public key cryptosystem- a tutorial “, 1998.
[11] A.Lewko and B.Waters, “Decentralizing attribute-based encryption”, Advances in Cryptology- EUROCRYPT 2011, pp. 568-588, 2011.

[12] A. Shamir, “How to share a secret”, Communications of the ACM, vol.22, no.11, pp.612-613, 1979.

[13] M.H.Dehkordi and S.Mashhadi, “New efficient and practical verifiable multi-secret sharing schemes”, Information Sciences, vol.178, no.9, pp.2262-2274, 2008.

[14] Z.Eslami and J.Z.Ahmadabadi “A verifiable multi-secret sharing scheme based on cellular automata”, Information Sciences, vol.180, no.15, pp.2889-2894, 2010.

[15] Chunqiang Hu, Wei Li, Xiuzhen Cheng,Jiguo Yu,Shengling Wang, Rongfang Bie, “ A Secure and Verifiable Access Control Scheme for Big Data Storage in Clouds”, IEEE Transactions on Big Data,DOI:10.1109/TBDATA.2016.2621106.

[16] S.V.Divya, R.S.Shaji, P.Venkadesh, “A Comprehensive Data Forwarding Technique under Cloud with Dynamic Notification”, Research Journal of Applied Sciences, Engineering and Technology, 7 (14), 2946-2953, 2014.