International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-9 Issue-1, May 2020

A Novel Cryptosystem for Files Stored in Cloud using NTRU Encryption Algorithm

N. Suba Rani, A. Noble Mary Juliet, S. Arunkumar

Abstract: ***The cloud refers to a set of services and infrastructure that are accessed via the internet. the cloud infrastructure is shared by many users each one performing different tasks. in order to prevent data leakage in the cloud the Cloud Service Provider should employ an Encryption Algorithm to protect the data of the users. Since the cloud service providers and large amount of data the Encryption Algorithm should be very efficient in terms of computational cost and time. Current Cloud service providers you son of the following algorithms to encrypt and decrypt the data Advanced encryption standard(AES), Rivest–Shamir–Adleman (RSA) and elliptic curve cryptography(ECC).The encryption for cloud should be chosen such that it is computationally efficient for the Cloud Service Provider and also meets the security requirement of the user***

Keywords: ***Advanced encryption standard (AES), Rivest–Shamir Adleman (RSA), Elliptic Curve Cryptography (ECC), NTRU Encryption.***

I. INTRODUCTION

A user or an organisation stores or outsourcers their data to another organisation which is called as Cloud Service Provider(CSP) in order to reduce initial investment for hardware and software required to setup their own network based storage. The user or the organisation which is outsourcing the data trusts the Cloud Service Provider and thinking CSP is honest and stores their data in a secure way without storing or looking into the operations on data. The process of securely storing data in cloud includes set of encryption algorithms and rules to be followed by the CSP and the user. But every cloud service provider should be treated as a honest but curious entity.

A. Symmetric Encryption:

Symmetric encryption is a group of encryption algorithms which has the simplest implementation of encryption algorithms. It uses of single key called a secret key to encrypt and decrypt data. The sender and also the receiver should have the Secret key in order to encrypt or decrypt the data. Usually the key will be generated by hashing the file and file will be encrypted using the hash key. The symmetric encryption algorithm like AES is mostly used in the cloud storages.

B. Asymmetric Encryption:

Unlike symmetric encryption asymmetric encryption uses two keys. One key for encryption and called the public key. The key is named public key because it is published publicly and anyone can view it. Another key for decryption and is called private key. The private key should be kept secret and in a secure manner. Because only the private key can decrypt the data. Algorithms like RSA, NTRU are based on

The asymmetric encryption techniques are more secure because they have two keys and of the two keys only one can decrypt the data which is stored securely and not exposed to the riskier environment by transmitting the key.

II. LIITRETURE REVIEW:

Nowadays all the organizations and even common people are moving towards cloud for storage and computational service. There are many advantages of using cloud storage, they are low initial investment for cloud systems and software, all the cloud service providers store the data in a redundant manner and in case of failure storage systems they can easily recover the data. The storage of data in cloud results in cloud-enabled collaboration between different users of the organization.

The present cloud service providers such as Google Cloud Platform, Amazon EC2, and Microsoft Azure use Advanced Encryption Standard (AES) Protecting data at rest. And use Rivest–Shamir–Adleman(RSA) for applications that involve passing of messages between two systems. All the Cloud Service Providers use AES-256 a 256-bit variant which has a 256 bit key for encrypting the data and has 14 rounds of calculations.[1][2].

However the above mentioned algorithms have some drawbacks. For any cryptographic algorithm it is considered an attack if someone can break the cipher text before a brute force attack. For AES there have been many number of instances where either key is recovered or it is attacked by side channel attacks also called as implementation attacks.

Manuscript received on April 02, 2020.
Revised Manuscript received on April 15, 2020.
Manuscript published on May 30, 2020.
* Correspondence Author
Dr. N. Suba Rani, Department of Computer Science and Engineering, Dr Mahalingam College of Engineering and Technology, Pollachi, India. E-mail: suba@drmcet.ac.in
Dr. A. Noble Mary Juliet, Department of Computer Science and Engineering, Dr Mahalingam College of Engineering and Technology, Pollachi, India. E-mail: cse.julie@drmcet.ac.in
S.Arunkumar, Department of Computer Science and Engineering, Dr Mahalingam College of Engineering and Technology, Pollachi, India. E-mail: s.arunkumar363@gmail.com

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

AES has an proven key recovery mechanism which can recover the key of the algorithm in a very less time of up to 1/3 to 1/5 of the actual time. The characteristics of the AES which allowed the attack possible are the design of AES round transformations which don't have resistance to many groups of attacks[5].

There are also some side channel attacks on AES which are based on the implementation of the AES. The attacker should have access to the machine which is computing the AES.

The attacker will monitor the cache memory and will modify a part of the cache memory. Then will check the amount of time taken to reload from the cache. This gives a frequency of the parts of the keys stored in the cache. With this information The AES key can be reconstructed in a time less than the time required for brute force attack.[6]

There are also some attacks on the RSA algorithm also, they are as follows. Whenever an instance of RSA is running there will always be errors in the computation. Those error values can be used to attack and find the ciphertext also RSA is vulnerable to index calculation attacks [7][8].

III. PROPOSED SYSTEM

In this paper we propose a novel approach for securing the files that are stored in the cloud using NTRU encryption algorithm. The NTRU is based on the algebraic properties of certain polynomial rings. The hard problem on which NTRU is based on is the shortest vector problem which is the finding a shortest vector in a lattice [9].

The NTRU is a public key cryptosystem which will have two keys namely public key and private key. The public key will be used foe encryption and the private key will be used for decryption.

The operations of NTRU takes place over a ring of truncated polynomials

$$p = Z_q[X]/(X^N - 1)$$

In the polynomial ring p a polynomial f is defined by

$$f = (f_0, f_1, f_2, f_{N-1})$$

$$f = f_0 + f1_x + f_{N-1}X^{N-1}$$

NTRU is made up of six publicly available integers

- N is prime and sufficiently large to prevent lattice attacks.
- p and q are relatively prime numbers.
- q is much larger than p.
- $L_f = B(d_f)$ is a set of polynomials where private keys are selected.
- $L_g = B(d_g)$ is a set of polynomials where public keys are selected

While choosing the parameters should be very careful to choose q much larger than p in order to prevent decryption failures.

A. Key Generation:

- Randomly choose a polynomial $f \in L_f$ such that f is invertible in modulo p and modulo q.
- Compute $f_p = f^{-1}$
- $f_q = f^{-1}(modulo\ q)$.
- Compute $h = g * f_q (mod\ q)$.
- Public key is (N, h).
- Private key is (f, f_p).

B. Ntru Encryption

The plain text should be converted into a polynomial format in order to do the mathematical operations on it.

- Convert the plaintext into a polynomial m such that $m \in L_m$.
- Choose a polynomial in random r where $r \in L_r$.

Now encrypt the plain text m using public keys(N, h) and using the encryption rule

$$e = p * r * h + m(mod\ q)$$

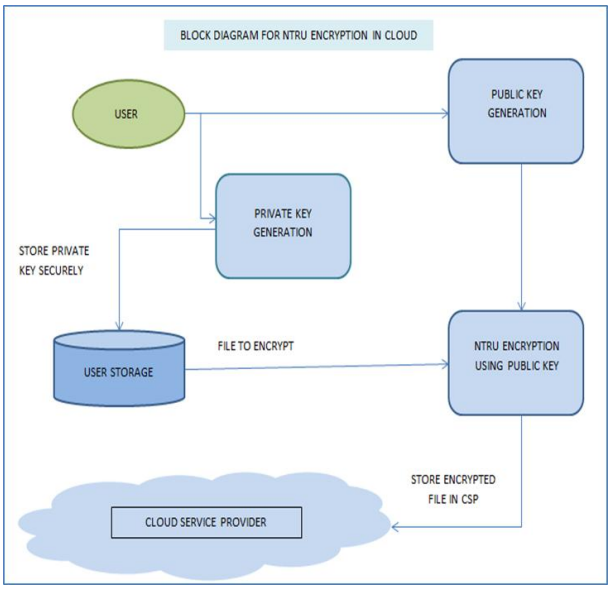

Figure 1 Block Diagram for Encryption in Proposed System

C. Ntru Decryption:

Compute $a = f * e\ (mod\ q)$.

- Transform a into a polynomial with coefficients in the interval $[-\frac{q}{2}, \frac{q}{2}]$ by using centring procedure.
- Compute $m = f_p * a\ (mod\ q)$

The computed m is the plain text after decryption.

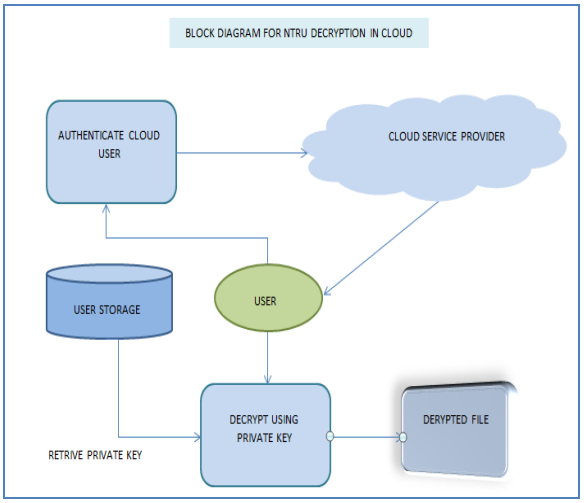

Figure 2 Block Diagram For Decryption In Proposed System

IV. RESULTS:

To evaluate the proposed system a system with an intel core i3 processor with 4gb of DDR3 RAM is used. Systems are given with Set of files of varying sizes ranging from 1 MB to 70MB. Both the systems are given with same set of files and the execution time is noted. The time taken to encrypt and decrypt the files are measured and taken as evaluation metrics.
The evaluation is done by using the following metrics such as,

- The time for encryption.
- The time for decryption.
- Total time taken for encryption and decryption.

A. The time taken for encryption:

The time taken to encrypt the files using both the algorithms are compared to evaluate the performance of the proposed system. The proposed system encrypts 75% faster than the existing system. This performance improvement is due to the fact that the existing system uses AES-256 to encrypt the files. The faster performance is the result of the simplicity in the implementation of the NTRU algorithm. The NTRU cryptosystem is fast because of the simple multiplication of the polynomials.

Table6.1 Time for encryption

TIME TAKEN FOR ENCRYPTION		
FILESIZE	Existing system	NTRU
4743	6	2
9496	6	3
18991	10	3
23739	10	3
28486	12	3
33234	15	4
37982	17	4
75963	28	8

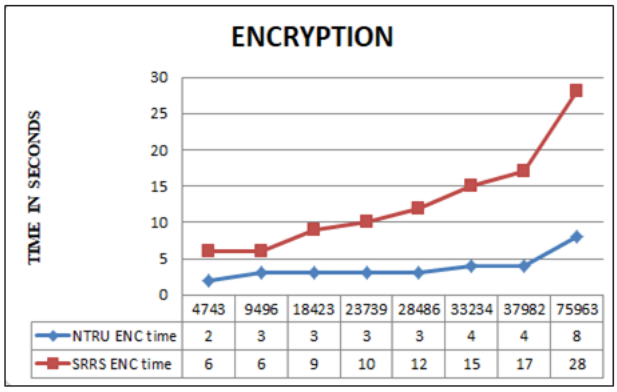

Figure 3 Time Taken For Encryption

B. The time taken for decryption:

The proposed NTRU cryptosystem offers a good 80% better performance compared to the existing system. The performance is due to the simple mechanism used to decrypt, finding the two shortest vectors in a given lattice. The proposed NTRU cryptosystem is very efficient in decrypting the data.

Table 6.2 Time for decryption

TIME TAKEN FOR DECRYPTION		
FILESIZE	Existing system (TIME-SEC)	NTRU (TIME-SEC)
4743	4	1
9496	6	1
18991	8	1.5
23739	10	2
28486	10	2.5
33234	11	3
37982	15	3
75963	20	4

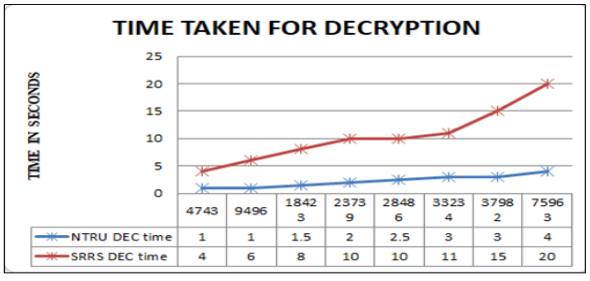

Figure 4 Time taken for decryption

C. The time taken for decryption:

The overall performance of the existing and proposed systems are measured by encrypting and decrypting simultaneously. The NTRU cryptosystem proved to be 73% faster than the existing system. The performance graph of the NTRU cryptosystem is almost linear when compared to the existing system. From the results we can conclude that the NTRU cryptosystem is very fast when compared to the existing systems. Also NTRU is very secure, which is not broken by anyone till this time.

Table 6.3 Overall Perfomance

TIME TAKEN TO ENCRYPT AND DECRYPT		
FILESIZE	NTRU	Existing system
4743	4	10
9496	4	12
18991	4	18
23739	5	20
28486	5	22
33234	6	26
37982	6	32
75963	12	48

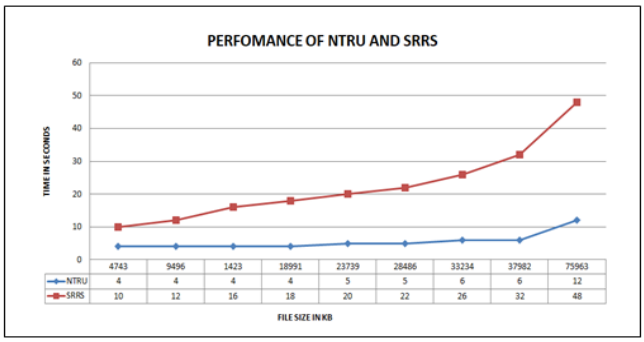

Figure 5 Overall performance

V. CONCLUSION

The NTRU cryptosystem proved to be 73% faster than the existing system. Thus a faster system means it will consume very less resources such as time and memory usage when compared to the existing system. NTRU also has a record of being not broken by anyone. NTRU is quantum safe which means there is no algorithm which can break it in a quantum time. Thus NTRU is the best algorithm suited for the security of files stored in the cloud systems.

REFERENCES

1. Nasarul Islam.K.V, Mohamed Riyas.K.V(2017),” Analysis of Various Encryption Algorithms in Cloud Computing”, IJCSMC,volume:6, pp 90-97.
2. Joan Daemen and Vincent Rijmen(2002), “The Design of Rijndael, AES - The Advanced Encryption Standard”, Springer-Verlag ,pp 238.
3. Ronald Linn Rivest,Adi Shamir,Leonard max Adleman(1983)”A method for obtaining digital signatures and public-key cryptosystems”,Communications of the ACM,Volume 26, Issue 1.
4. Acıiçmez O., Schindler W., Koç Ç.K. (2006) Cache Based Remote Timing Attack on the AES. In: Abe M. (eds) Topics in Cryptology – CT-RSA 2007. CT-RSA 2007. Lecture Notes in Computer Science, vol 4377. Springer, Berlin, Heidelbe.
5. Bogdanov A., Khovratovich D., Rechberger C. (2011) Biclique Cryptanalysis of the Full AES. In: Lee D.H., Wang X. (eds) Advances in Cryptology – ASIACRYPT 2011. ASIACRYPT 2011. Lecture Notes in Computer Science, vol 7073. Springer, Berlin, Heidelberg.
6. Bonneau J., Mironov I. (2006) Cache-Collision Timing Attacks Against AES. In: Goubin L., Matsui M. (eds) Cryptographic Hardware and Embedded Systems - CHES 2006. CHES 2006. Lecture Notes in Computer Science, vol 4249. Springer, Berlin, Heidelberg.
7. Aumüller C., Bier P., Fischer W., Hofreiter P., Seifert JP. (2003) Fault Attacks on RSA with CRT: Concrete Results and Practical Countermeasures. In: Kaliski B.S., Koç .K., Paar C. (eds) Cryptographic Hardware and Embedded Systems - CHES 2002. CHES 2002. Lecture Notes in Computer Science, vol 2523. Springer, Berlin, Heidelberg.
8. Coron, JS., Naccache, D., Desmedt, Y. et al. Des Codes Crypt (2006) 38: 41. doi.org/10.1007/s10623-004-5660-y.
9. Hoffstein J., Pipher J., Silverman J.H. (1998) NTRU: A ring-based public key cryptosystem. In: Buhler J.P. (eds) Algorithmic Number Theory. ANTS 1998. Lecture Notes in Computer Science, vol 1423. Springer, Berlin, Heidelberg

AUTHORS PROFILE

Dr. N. Suba Rani completed her doctorate in Information and Communication Engineering from Anna University in the year 2017. She is currently working as Assistant professor at Dr.Mahalingam College of Engineering and Technology, Tamilnadu. Her areas of Interest are Cloud Computing, Network security, Machine Learning and Internet of Things.

Dr. A. Noble Mary Juliet is an Associate professor in Computer Science and Engineering at Dr.Mahalingam College of Engineering and Technology. She received her ME degree in Computer Science and Engineering from Anna University, in 2006, and her Ph.D. degree in Information and Communication Engineering from Anna University, in 2016. Her research interest includes mobile and wireless computing, Networks and Internet of Things. She involves in many research projects and she is a reviewer for scientific journals and conferences.

Arunkumar. S PG Scholar, Department of Computer Science and Engineering, Dr.Mahalingam College of Engineering and Technology(MCET), Pollachi.He has completed his Bachelor of Engineering in sri shakthi institute of engineering and technology, Coimbatore.He is doing his project in cloud security.