International Journal of Pure and Applied Mathematics
Volume 117 No. 15 2017, 675-681
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

ijpam.eu

ENHANCED HOMOMORPHIC ENCRYPTION TECHNIQUE IN CLOUD COMPUTING

[1]M.Martinaa, [2]E.Ponmani, [3]G.Bagyalakshmi, [4]S.AnanthaKrishnan

[1,2,4]School of Computing, SASTRA UNIVERSITY, Thanjavur – 613402, Tamil Nadu, India

[3]School of EEE, SASTRA UNIVERSITY, Thanjavur – 613402, Tamil Nadu, India

ABSTRACT

Cloud Computing is a framework of computing, that comprises of all components like software, hardware, and networking that are essential to empower evolution of cloud services across the Internet. As the entire data processing in cloud computing is done across the internet and private networks, it may be vulnerable to many attacks and prone to threats, there is a requirement to protect it. It is a well known fact that data in cloud must be protected through encryption and the encrypted data should be decrypted during data processing. Homomorphic Encryption techniques enable data processing of encrypted data and there-by reduces latency and computational complexity. This paper focuses on enhancing the security in cloud computing through the usage of homomorphic encryption technique.

Keywords: Homomorphic encryption, Paillier cryptosystems, Proxy re-encryption algorithm

1. INTRODUCTION

Today, in the world of cloud computing[9], where we get each and every requirement of ours on demand, and every requirement of an individual is met by the cloud services. Cloud computing is a model of computing, that refers to access one's needs as per the requirement.

Encryption refers to a method of encoding messages or information in a way that only the authorized parties can have access to it. Homomorphic Encryption is an encryption technique, that enables the processing of encrypted data, that is it allows simple operations like string concatenation, addition, subtraction, etc to be performed directly on the encrypted text(cipher text) without having the need to decrypt it, there by the sender need not share details about the encryption key. These homomorphic encryption systems are malleable. Since they are malleable, they can be used in cloud computing environment to process private data without losing its confidentiality. A homomorphic encryption has the following properties:

1. Additive.
$$F(x \oplus y) = F(x) \oplus F(y)$$
2. Multiplicative.
$$\mathrm{F}(\mathrm{x} \otimes \mathrm{y}) = \mathrm{F}(\mathrm{x}) \otimes \mathrm{F}(\mathrm{y})$$

Where F(x) = Encryption(x)

The keys are generated by the client. The public key is acquired by the client. Encrypt the data in a homomorphic encryption scheme using private key.Next we obtain a random cipher data generated by a random key on passing the encrypted data to the proxy re-encryption algorithm.

2. LITERATURE SURVEY

This paper is proposed to present homomorphic encryption schemes based on proxy re-encryption algorithm, paillier systems and el gamal cryptosystems.In Proxy re-encryption schemesproxiescan alter a ciphertext such a way that it could be decrypted by one party while it has been encrypted for another. This enables the sender, say A to reveal the contents of messages that are sent to C, a third party. Along-side A maintains the confidentiality of his key and doesn't reveal his private key to C. More over the proxy is not allowed to read the contents of A's messages. Thereby A assigns or selects a proxy to carry few operations like re-encrypting the messages that are sent to C. This results in generation of a new key that could beused by C for decryptingthe message later. These proxies have the ability toalter the message, allowing its decryption by C. This method can be applied in numerous applications like, content distribution, e-mail forwarding and law-enforcement monitoring.

Pailliercryptosystems, the example of a probabilistic asymmetric algorithm, is a homomorphic cryptosystem. It is an additive homomorphic system, which means the encryption of (m1 + m2) can be computed without decrypting the individual messages. Here m refers to any random message.ElGamal encryption is defined over a cyclic group G.Its principle ismainly based on the Diffie–Hellman key exchange and its security depends upon the problem in G,associated to computing discrete logarithms.

MS. Parin V. Patel[1] proposed a homomorphic encryption to assure security on cloud. In these systems only the client holds private key and it is used to carry out different operations on encrypted data without decryption. Alecsandru Patrascu[2]proposed a schemeon new directions in cloud computing, which refers to availability,performance, and security.Jing-Li Han[3] gave a description on the developments in Fully Homomorphic Encryption.It is a special type of encryption system that permits arbitrarily complex computation on encrypted data.

Nitin Jain[4] described how Gentry's transformation can be applied on boot-strappable Somewhat Homomorphic Encryption (SHE) scheme to make it a Fully Homomorphic scheme (FHE) by squashing the decryption circuit which means evaluating with a low-degree polynomial.Vinod Vaikuntanathan[5] proposed a paperproviding a survey focusing on the development of Homomorphic Encryption to FullyHomomorphic Encryption.He also discussed about the applications of FullyHomomorphic Encryption.

3. PROPOSED WORK

Previous works were manly published on proxy re-encryption algorithm based on paillier systems and RSA Cryptography in order to prevent Chosen Cipher Text Attack on the Cipher Text. This proposal is presented on proxy re-encryption algorithm in accordance withElGamal crypto systems. Here, the client encrypts the data using a set of keys called the secretkey and the public key, where secret key is the private key kept with the client. The encrypted data is sent to proxy re-encryption algorithm to obtain random key generated cipher data. Thus even on obtaining the random generated key, the intruder may have to decrypt the data twice for which he may require two keys. Thus it can withstand the attacks.

Even when the key is randomly generated each and every time it is not possible to decrypt very easily and there -by it enhances the security.

3.1 System Architecture

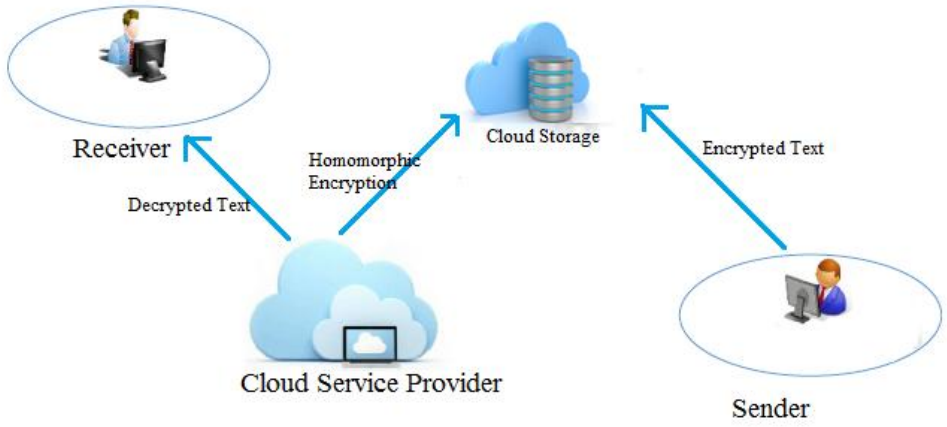

Figure 1: System Architecture

Figure 1, depicts the System Architecture of the proposed algorithm. It comprises four types of entities,

i. Sender.
ii. Receiver.
iii. Cloud Service Provider.
iv. Cloud Storage.

The plain text is encrypted by the sender (client) and then they obtained cipher text is stored in cloud storage which is accessed by the cloud service provider, who will then re encrypt the data with the proposed algorithm. If the client requests any operations to be performed, the cloud service provider performs them and on demand the client can decrypt the data as required.

The design of the algorithm follows a flow as follows:

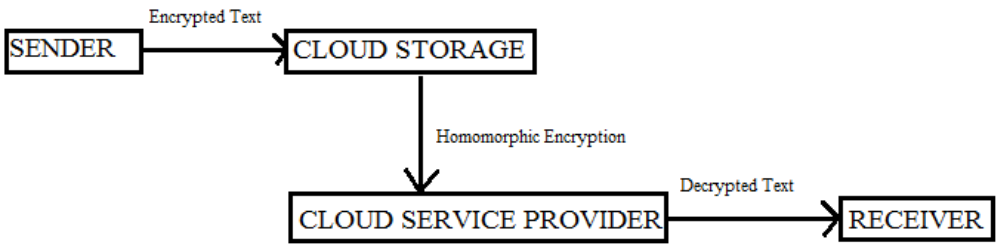

Figure 2. Flow Diagram

Figure 2, depicts the flow diagram which can be depicted as; the client first sends Encrypted data to the cloud service provider, who then re encrypts the data with the proposed algorithm. Thus when the client requests for any operations, they are performed on the cloud and whenever needed the client decrypts the data.

3.2 PROCEDURE

The new el gamal basedproxy re encryption have the following stages:

1. Key Generation:

   P belongs to **P**, alpha is a primitive root of p
   where**P**is a set of Prime Numbers.
   Select a random integer X_A, such that $1< X_A< (p-1)$
   Y_A=(exp (alpha,X_A)) modulo p
   Public Key = {p, alpha, Y_A}
   Secret Keyor Private Key X_A.

2. Encryption:

   Plain Text M<p
   Select k, k<p where k is a random integer.
   K= (exp (Y_A, k)) modulo p
   C1= (exp (alpha, k)) modulo p
   C2= KM mod p
   Cipher Text: (C1, C2)

3. Re Encryption:

   Compute the Private Key Rpk and Secret Key Rsk
   Re-encrypt the cipher text generated by ElGamal and send Rpk to cloud server.

4. Decryption:

   Cipher Text: (C1, C2)
   K= exp (C1, XA) modulo p
   M=($C2K^{-1}$) modulo p.

   where exp (x, y) = x ^ y.

Table.1 Proposed Proxy Re Encryption based on ElGamal Cryptosystems.

Key Generation	Encryption	Proxy – Re Encryption	Decryption
p belongs to P, alpha is a primitive root of p where P is a set of Prime Numbers.	Plain Text M<p	Compute the Private Key Rpk and Secret Key Rsk	Cipher Text: (C1, C2)
Select a random integer X_A, such that $1< X_A <(p-1)$	Select a random integer k, k<p	Re encrypt the cipher text generated by ElGamal and send Rpk to cloud server.	K= exp (C1, X_A) modulo p $M=(C2K^{-1})$ modulo p
Y_A = exp(alpha, XA) modulo p	K= exp (Y_A,k) modulo p		
Public Key PU={p, alpha, Y_A}	C1=exp(alpha, k) modulo p C2= KM modulo p		
Secret Key X_A	Cipher Text: (C1,C2)		

4. Conclusion

Previous works quoted the usage of RSA Algorithm to develop the proxy re-encryption algorithm. This paper ispresented based on elgamal crypto systems. ElGamal Cryptosystems is advantageous over RSA algorithm in which ElGamal is faster in terms of encryption and decryption process and also ElGamal consumes lesser power compared to RSA algorithm.Thus, the proposed Proxy Re Encryption algorithm based on ElGamal cryptosystems ensures a better encryption technique.

5. References

1. Ms. Parin V. Patel, Mr. Hitesh D. Patel, Prof. Pinal J. Patel, A Secure Cloud using Homomorphic Encryption Scheme, International Journal of Computer Science Research & Technology (IJCSRT) Vol. 1 Issue 1, June-2013.

2. AlecsandruPatrascu, Diana Maimu, Emil Simion, New Directions in Cloud Computing: A Security Perspective, IEEE 2012.

3. Jing-Li Han, Ming Yang, Cai-Ling Wang, Shan-Shan Xu, TheImplementation and Application of Fully Homomorphic Encryption Scheme,IEEE 2012.

4. Nitin Jain, SaibalK. Pal &Dhananjay K. Upadhyay, Implementation andAnalysis of Homomorphic Encryption Schemes, International Journal on Cryptography and Information Security(IJCIS), Vol.2, No.2, June 2012.

5. VinodVaikuntanathan, Rajeev Shukla,Computing Blindfolded: NewDevelopmentsin Fully Homomorphic Encryption, International Journal ofComputer Science Research &Technology (IJCSRT), June- 2011.

6. Ji-Jiang,Yangab, Jian-Qiang Li c, Yu Niub, A hybrid solution for privacy preserving medical data sharing in the cloud environment , Future Generation Computer Systems, Elsevier 2015.

7. Andrei Ciocan, SergiuCostea, Implementation and optimization of a Somewhat Homomorphic Encryption Scheme, IEEE 2015.

8. JungHeeCheon, HyunsookHong, MoonSungLee, HansolRyu, The polynomial approximate common divisor problem and its application to the fully homomorphic encryption, Information Sciences, Elsevier 2016.

9. Dr.AnanthiSheshasaayee and R. Megala, "A Conceptual Framework For Resource Utilization In Cloud Using Mapreduce Scheduler", International Journal of Innovations in Scientific and Engineering Research (IJISER), Vol.4,no.6,pp.188-190, 2017.
10. Rajesh, M., and J. M. Gnanasekar. "Annoyed Realm Outlook Taxonomy Using Twin Transfer Learning." International Journal of Pure and Applied Mathematics 116 (2017): 547-558.
11. Rajesh, M. & Gnanasekar, J.M. Wireless Pers Commun (2017),https://doi.org/10.1007/s11277-017-4565-9
12. Rajesh, M., and J. M. Gnanasekar. "GCCover Heterogeneous Wireless Adhoc Networks." Journal of Chemical and Pharmaceutical Sciences (2015): 195-200.
13. Rajesh, M., and J. M. Gnanasekar. "CONGESTION CONTROL IN HETEROGENEOUS WANET USING FRCC." Journal of Chemical and Pharmaceutical Sciences ISSN 974: 2115.
14. Rajesh, M., and J. M. Gnanasekar. "GCCover Heterogeneous Wireless Ad hoc Networks." Journal of Chemical and Pharmaceutical Sciences (2015): 195-200.
15. Rajesh, M., and J. M. Gnanasekar. "CONGESTION CONTROL USING AODV PROTOCOL SCHEME FOR WIRELESS AD-HOC NETWORK." Advances in Computer Science and Engineering 16.1/2 (2016): 19.
16. RAJESH, M. "TRADITIONAL COURSES INTO ONLINE MOVING STRATEGY." The Online Journal of Distance Education and e-Learning 4.4 (2016).
17. Rajesh, M. "Object-Oriented Programming and Parallelism."
18. Rajesh, M., K. Balasubramaniaswamy, and S. Aravindh. "MEBCK from Web using NLP Techniques." Computer Engineering and Intelligent Systems 6.8: 24-26.