A Secure Cloud Using Homomorphic Encryption

[1]Prof. Parin Patel, [2]Prof. Hitesh Patel, [3]Prof Kiran Patel
Assistant Professor
Gandhinagar Institute of Technology, Gandhinagar, India

***Abstract*— Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. There are many problem related with cloud computing traffic, security and resource management. We can provide security in cloud by many ways like on data, network and storage. I propose homomorphic encryption to provide security on cloud. Homomorphic Encryption systems are used to perform operations on encrypted data without knowing the private key (without decryption), the client is the only holder of the secret key. When we decrypt the result of any operation, it is the same as if we had carried out the calculation on the raw data. This method provides more security on data because provider is not involving in key management. I use proxy re-encryption technique that prevents ciphertext from chosen cipher text attack. This system is more secure than existing system.**

***IndexTerms*— Cloud Computing, security, Homomorphic Encryption, RSA, Pallier**

I. INTRODUCTION

A. What is cloud computing?

The definition of cloud computing provided by National Institute of Standards and Technology (NIST) says that: "Cloud computing is a model for enabling convenient, on- demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. [2]".

Cloud computing provides better utilization of distributed resources over a large data and they can access remotely through the internet. Cloud computing have many advantages on cost and other side.

Cloud Computing Architecture

Cloud computing system is divided into two sections: the front end and the back end. Front end through which user can interact with the server and backend is the server which provides data to the client. Between server and client network is working as middleware.

- Layers and Services of Cloud Computing Architecture
  - (Cloud) Infrastructures as a Service (IaaS) provide data centers, servers and other resources. There is no need to create any infrastructure for cloud.
  - (Cloud) Platform as a Service (PaaS) is provides computational resources via a platform upon which services and application can be developed and hosted. Example: Google Docs, SAP business by design.
  - (Clouds) Software as a Service (SaaS) is also sometimes referred to as Services or application clouds.
- Deployment of Cloud Computing Service

  Three types of deployment services- public cloud, private cloud and hybrid cloud:
  - Public cloud allows the user to access cloud via network. This cloud is publicly available on internet so security is the big problem. This cloud is on “Pay and Use” basis.
  - Private Cloud is within an organization. This stores the internal data of organization. It is more secure and maintenance is also easy.
  - The Hybrid Cloud is a combination of any cloud services. Community cloud is constructed by many organizations according to their requirements.

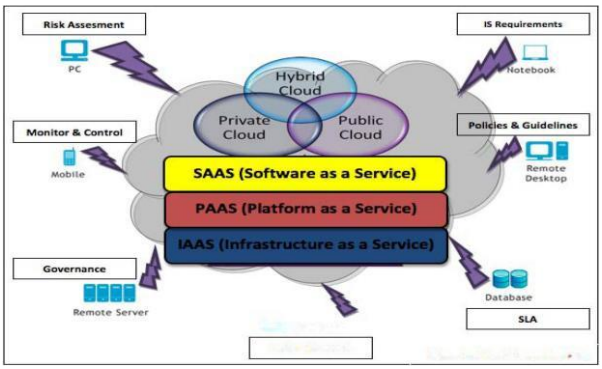

Fig. 1 Cloud Architecture

Advantages and Disadvantages of Cloud Computing

Advantages:

- Easy to Maintain.
- Less cost.
- We can use on pay and use basis.
- Personalized Backup and recovery.
- Remote access.
- Green computing.

Disadvantages:

- Security and privacy.
- Higher operational cost.

B. Homomorphic Encryption

Homomorphic encryption alludes to encryption where plain texts and cipher texts both are treated with an equivalent algebraic function. Homomorphic Encryption allows server to do operation on encrypted data without knowing the original plaintext.

Homomorphic encryption allows complex mathematical operations to be performed on encrypted data without using the original data. For plaintexts X1 and X2 and corresponding ciphertext Y1 and Y2, a Homomorphic encryption scheme permits the computation of X1 Θ X2 from Y1 and Y2 without using P1 Θ P2.The cryptosystem is multiplicative or additive Homomorphic depending upon the operation Θ which can be multiplication or Security is biggest problem of cloud computing. Many Research paper discuss about cloud and it's advantage and disadvantage. In my Literature review I found security is major key point. From the Literature Review I found Homomorphic encryption is the more secure encryption scheme. In this scheme cloud server can perform any algebraic operation on cipher data. From literature Review I found that Chosen Cipher text attack is major problem.

II. EXISTING SYSTEM

A Homomorphic encryption has different Homomorphic schemes according to its properties:

Security is biggest problem of cloud computing. Many Research paper discuss about cloud and it's advantage and disadvantage. In my Literature review I found security is major key point. From the Literature Review I found Homomorphic encryption is the more secure encryption scheme. In this scheme cloud server can perform any algebraic operation on cipher data. From literature Review I found that Chosen Cipher text attack is major problem.

A. Additive Homomorphic Encryption:

In additive Homomorphic encryption sum of encrypted cipher text is same as sum of original plain text. This property allows you to apply addition on encrypted data without knowing original data.

A Homomorphic encryption is additive, if: [10]

$Enc(x \oplus y) = Enc(x) \otimes Enc(y)$

$Enc(\Sigma\, mi) = \Pi\, Enc(mi)$

$i=1 \quad i=1$

Suppose we have two ciphers C1 et C2 such that:

$C1 = g_{m1} .r1_n \bmod n_2$

$C2 = g_{m2}. r2_n \bmod n_2$

$C1.C2 = gm1.r1_n.g_{m2}.r2_n \bmod n_2 = g_{m1+m2} (r1r2)_n \bmod n_2$

So, Pailler, Benaloh and Okamoto-Uchiyama cryptosystems realizes the property of additive Homomorphic encryption. An application of an additive Homomorphic encryption is electronic voting: Each vote is encrypted but only the "sum" is decrypted.

Cloud server contains encrypted vote it just perform addition on encrypted data and get the encrypted result. That encrypted result is decrypted at the client side and get the original result.

B. Multiplicative Homomorphic Encryption
In Multiplicative Homomorphic encryption Multiplication of encrypted cipher text is same as Multiplication of original plain text. This property allows you to apply Multiplication on encrypted data without knowing original data.
A Homomorphic encryption is multiplicative, if: [10]

Enc (x⊗y) = Enc(x) ⊗ Enc(y)
1 l
Enc (Πmi) = Π Enc (mi)
i=1 i=1

RSA and Elgamal cryptosystems realize the properties of the multiplicative Homomorphic encryption.

III. PROBLEM IN EXISTING SYSTEM

Suppose we have two ciphers C1 et C2 such that:
$C1 = m1_e \bmod n$
$C2 = m2_e \bmod n$
$C1.C2 = m1em2e \bmod n = (m1m2)e \bmod n$
RSA cryptosystem is working with property of multiplicative Homomorphic encryption, but it has a lake of security, because if we have two ciphers C1, C2 corresponding respectively to the messages m1,m2 so:
$C1 = m1_e \bmod n$
$C2 = m2_e \bmod n$

The client sends the pair (C1, C2) to the Cloud server and server performs the calculations requested by the client and sends the encrypted result (C1 × C2) to the client.

If the attacker intercepts two ciphers C1 et C2, which are encrypted with the same private key, so they are able to decrypt all messages exchanged between the server and the client. Because the Homomorphic encryption is multiplicative, i.e. the product of the ciphers equals the cipher of the product.

The basic RSA algorithm and Paillier Cryptosystem is vulnerable to chosen ciphertext attack (CCA).CCA is defined as an attack in which adversary chooses a number of ciphertext and is given the corresponding plaintext, decrypted with the target's private key. Thus the adversary could select a plaintext, encrypt it with the target's public key and then be able to get plaintext back by having it decrypted by private key. So attacker will know the entire data in-between client and cloud server.

IV. PROPOSED SYSTEM

To prevent cipherdata from CCA (chosen ciphertext attack) I propose Proxy Re-Encryption algorithm with paillier and RSA Cryptosystem. In Homomorphic encryption scheme data was encrypted by the private key and publickey was kept with client only. We again pass that data in proxy re-encryption algorithm and get every time random key generated cipherdata. If attacker gets that key ones then they need to decrypt that data twice with two different keys. If once attacker gets the plaintext than he is not able to get every plaintext between client and server. So this system provides more security than existing system.

A. Proxy Re-encryption Algorithm
Proxy Re-Encryption (CiphertText)
Key Generation -keygen(p,q)
1. Take two prime number p and q.
2. Compute n=p.q, Φ(n)=(p-1)(q-1) and choose e such that gcd(e, Φ(n)))=1.
3. Determine d such that e.d=1 mod Φ(n).
4. The Proxy public key (Rpk) is (e,n) is generated.
5. The proxy Secret key (Rsk) is (d) is generated.

Encryption: Enc (c,Rpk)
1. Let m be a message to be encrypted where m Є Zn.
2. Compute ciphertext as: rc=me mod n.

Decryption: Dec (rc,Rsk)
1. Ciphertext c Є Zn.
2. Compute message m=cd mod n.

B. Proposed Proxy Re-Encryption Based Paillier Algorithm:

Key generation:
1. Choose two large prime numbers p and q randomly and independently of each other such that gcd (pq ,(p-1)(q-1))=1.
2. Compute n=pq and λ=lcm (p-1, q-1).

3. Select random integer g where g Є Z*n2
4. Ensure n divides the order of g by checking the existence of the following modular multiplicative inverse: μ=(L(a λ mod n2))-1 mod n, where function is defined as L(u)=u-1/n.
5. The public (encryption) key is. (n,g)
6. The private (decryption) key is (λ, μ)

Encryption: Enc (m, pk)
1. Let m be a message to be encrypted where m Є Zn.
2. Select random where r Є Zn*.
3. Compute ciphertext as: $c=g_m . r_n \bmod n_2$.

Proxy Re-Encryption(c)
1. Compute Private and Public key.(Rsk,Rpk).
2. Re Encrypt Ciphertext generated by Paillier algorithm and send Public key (Rpk) to cloud server.

Decryption: Dec(c,sk)
1. Ciphertext c Є Z_{n2*}.
2. Compute message: m=L(c λ mod n2)/ L (g λ mod n2). Mod n

C. Proposed proxy Re-Encryption based RSA algorithm:

Key Generation -keygen(p,q)
1. Take two prime number p and q.
2. Compute n=p.q, Φ (n)=(p-1)(q-1) and choose e such that gcd(e, Φ(n)))=1.
3. Determine d such that e.d=1 mod Φ(n).
4. The public key (pk) is (e,n) is generated.
5. The Secret key (sk) is (d) is generated.

Encryption: Enc (m,pk)
1. Let m be a message to be encrypted where m Є Zn.
2. Compute ciphertext as: $c=m_e \bmod n$.

Proxy Re-Encryption(c)
1. Compute Private and Public key.(Rsk,Rpk)
2. Re Encrypt Ciphertext generated by Paillier algorithm and send Public key (Rpk) to cloud server.

Decryption: Dec(c,sk)
1. Ciphertext c Є Zn.
2. Compute message $m=c_d \bmod n$.

V. CONCLUSION

In this paper I have use Homomorphic encryption technique to provide security on cloud. Homomorphic encryption is a new concept of security which enables providing results of calculations on encrypted data without knowing the raw data on which the calculation was carried out, with respect of the data confidentiality. In this paper I have proposed RSA and Paillier algorithm for homomorphic encryption using proxy-Re-encryption algorithm that prevents cipher data from Chosen Cipher text Attack (CCA).So This system is more secure than existing system. In future we can optimize more efficiency of the system by reducing size of the key and we can also check proxy Re-Encryption method for other Homomorphic Encryption Scheme.

REFERENCES

[1] John Harauz, Lorti M. Kaufinan. Bruce Potter, "Data Security in the World of Cloud Computing", IEEE Security & Privacy, Copublished by the IEEE Computer and Reliability Societies, July/August 2009.
[2] National Institute of Standards and Technology- Computer Security Resource Center -www.csrc.nist.gov
[3] Cloud Computing http://en.wikipedia.org/wiki/Cloud_computing
[4] Yashpalsinh Jadeja and Kirit Modi, "Cloud Computing - Concepts, Architecture and Challenges", International Conference on Computing, Electronics and Electrical Technologies [ICCEET], IEEE-2012
[5] Samerjeet kaur, "Cryptography and Encryption in Cloud Computing", VSRD International Journal of Computer Science and Information Technology, VSRD-IJCSIT, Vol. 2 (3), 2012
[6] Ramgovind S, Eloff MM, Smith E, 'The management of security in cloud computing", IEEE – 2010
[7] Aderemi A. Atayero and Oluwaseyi Feyisetan," Security Issues in Cloud Computing: The Potentials of Homomorphic Encryption" Journal of Emerging Trends in Computing and Information Sciences, VOL. 2, NO. 10, October 2011
[8] Caroline Fontaine and Fabien Galand,"A Survey of Homomorphic Encryption for Nonspecialists",EURASIP Journal on Information Security, pages 1 to15, January 2007.

[9] Jibang Liu, Yung-Hsiang Lu and Cheng-Kok Koh," Performance Analysis of Arithmetic Operations in Homomorphic Encryption", ECE Technical Reports Paper 404, 2010
[10] Nitin Jain, Saibal K. Pal & Dhananjay K. Upadhyay." Implementation And Analysis Of Homomorphic Encryption Schemes" International Journal on Cryptography and Information Security (IJCIS), Vol.2, No.2, June 2012
[11] Kerckhoffs, A., (1883). "La cryptographie militaire (part i)", Journal des Sciences Militaires, Vol. 9, no. 1, pp. 5–38.
[12] Kerckhoffs, A., (1883). "La cryptographie militaire (part ii)", Journal des Sciences Militaires, Vol. 9, no. 2, pp. 161–191.
[13] Cloud Computing http://en.wikipedia.org/wiki/Cloud Computing
[14] Patrick Schmidt., (2011), Fully Homomorphic Encryption: Overview and Cryptanalysis, Diploma Thesis, Technische Universität Dortmund.
[15] Maha TEBAA, Saïd EL HAJJI and Abdellatif EL HAZI,"Homomorphic Encryption Applied to the Cloud Computing Security", Proceedings of the World Congress on Engineering, Vol I, London, U.K. July 4 - 6, 2012
[16] Mahadevan Gomathisankaran, Akhilesh Tyagi, Kamesh Namuduri, HORNS: A Homomorphic Encryption Scheme for Cloud Computing using Residue Number System",IEEE, 12 May 2011
[17] Youssef Gahi, Mouhcine Guennoun, Khalil El-Khatib," A Secure Database System using Homomorphic Encryption Schemes", The Third International Conference on Advances in Databases, Knowledge, and Data Applications, 2011
[18] ShahadiFarah,M.YounasJaved,Azara Shamim,Tabbassam Nawaz,"An Experimental study on performance Evaluation of Asymmetric Algorithms",Recent Advances in Information Science ISBN: 978-1- 61804-140-1-2012