A Survey on Homomorphic Encryption in Cloud Computing

[1]Rana M Pir, [2]Rumel M S Pir, [3]Imtiaz U Ahmed

[1]Lecturer, [2]Assistant Professor, [3]Lecturer
[1]Leading University, Sylhet Bangladesh,
[2] Leading University, Sylhet Bangladesh,
[3]Cox's bazar International University Bangladesh

***Abstract*—Cloud Computing is the delivery of service rather than product. You can share information to the clients on the cloud. Cloud provider can be of service provider, data provider and platform provider. There is major and challenging issue of cloud like security. The advantage of cloud is reduced cost, easy maintenance and providing of resources. In this paper I have survey the approach of Homomorphic encryption in cloud. Homomorphic encryption is the technique through which we can apply the algebraic operation on ciphertext without converting it in plaintext. Homomorphic encryption is the method which performs operation on encrypted data which will provide result without decrypting that data. This method provides the same result as operation performs on row data. The scheme is said to be "fully Homomorphic'' when we can perform (a sequence of operations) both addition and multiplication, whereas, it is "somewhat Homomorphic'' if it supports a limited number of operations.**

***Index Terms*—Homomorphic Encryption, Cloud Computing, Plaintext, Security**

I. INTRODUCTION

The word "cloud" originates from the world of telecommunications when providers start to using virtual private network (VPN) services for communications. 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. [3]". The history of the term cloud is originates from the world of telecommunications when providers start to using virtual private network (VPN) services for communications with comparable quality of service at lower cost. Initially they are using point-to-point data circuits which have wastage of bandwidth. But by using VPN services, they can balance the traffic. Cloud computing now covers servers and network infrastructure. So through this cloud computing there is no need to store the data on desktops, portables etc .You can store the data on servers and you can access the data through internet. Cloud computing provides better utilization of distributed resources over a large data and they can access remotely through the internet.

II. CLOUD ARCHITECTURE

Cloud computing is divided in two section: front end(to interact with user) and back end(data provided by server).There are three types of services provided by cloud:

1. Platform as a Service (PaaS): This service provides platform on which application can build, test and deploy. There is no need to buy any software. You need to pay for the time that use.
2. Infrastructure as a Service (IaaS): Data Server and Software, processors and other environment are also provided by this service. The whole infrastructure that you will get in this service so there is reduction of cost.
3. Software as a Service (SaaS): This service provides online software that you can use. So there is no need to install any software on your system.

Cloud has three different types of Deployment Services:

1. Public cloud allows the user to access cloud via network using browsers. This cloud is publicly available on internet so security is the big problem. In this cloud up gradation and maintenance is difficult. This cloud is on "Pay and Use" basis. You need to pay only the time duration that you have use.
2. The Hybrid Cloud is a combination of private and public cloud. When there is a need private cloud can links to one or more external services. Application and data are accessed through authentication.
3. Community cloud is constructed by many organizations according to their requirements. Cloud infrastructure is managed by third party or one of the organizations.

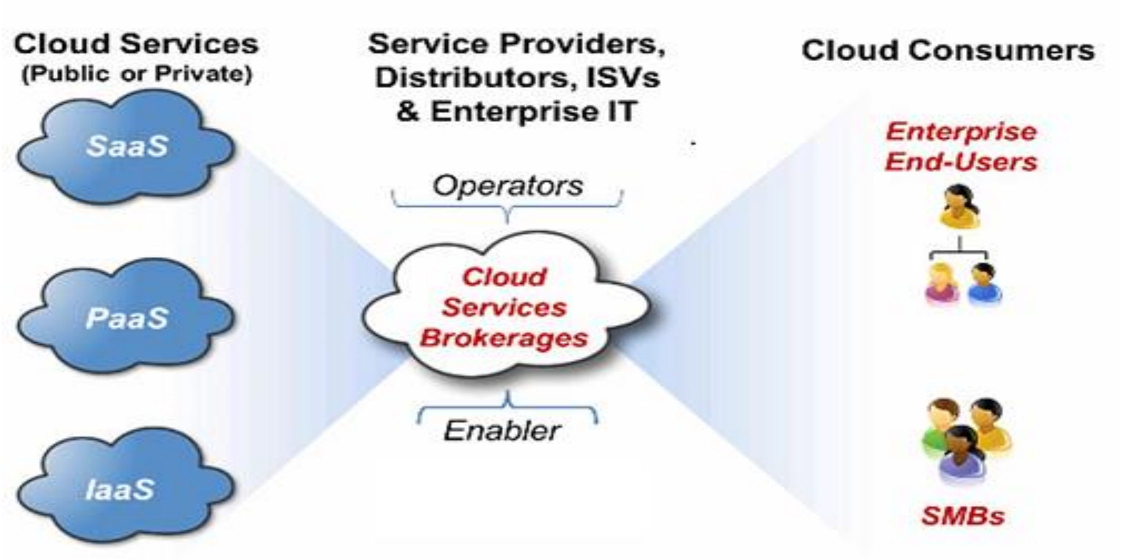

Figure 1 Cloud Computing Architecture [2]

III. ADVANTAGES AND DISADVANTAGES

Advantages:

1. Easy to Maintain.
2. Less cost.
3. Personalized Backup and recovery.
4. We can use on pay and use basis.
5. Green computing.
6. Remote access.

Disadvantages:

1. Higher operational cost.
2. Security and privacy.

IV. INTRODUCTION TO HOMOMORPHIC ENCRYPTION

Homomorphic encryption alludes to encryption where plain texts and cipher texts both are treated with an equivalent algebraic function. So the plain text and cipher text might not be related but emphasis on the algebraic operation that works on both.

Structured Encryption: A structured encryption scheme encrypts structured data with the using of secret keys. In addition, the query process reveals no useful information about either the query or the data. An important consideration is the efficiency of the query operation on the server.

Homomorphic encryption allows server to do the operations on encrypted data without access to the original data. Figure 3 shows the general framework. Homomorphic encryption is considered too expensive. Among the Homomorphic encryption we distinguish, according to the operations that allows to assess on raw data, the additive Homomorphic encryption (only additions of the raw data) is the Pailler and Goldwasser-Micalli cryptosystems, and the multiplicative Homomorphic encryption (only products on raw data) is the RSA and El Gamal cryptosystems.

1. Additive Homomorphic Encryption
   A Homomorphic encryption is additive, if: [1]
   $\mathrm{Enc}\ (x \oplus y) = \mathrm{Enc}(x) \otimes \mathrm{Enc}(y)$
   1 l
   $\mathrm{Enc}\ (\Sigma\ mi) = \Pi\ \mathrm{Enc}\ (mi)$
   i=1 i=1
   Example: Paillier Cryptosystem (1999):
   Suppose we have two ciphers C1 et C2 such that:
   C1 = gm1 .r1n mod n2
   C2 = gm2. r2n mod n2
   C1.C 2 = gm1.r1n.gm2.r2nmod n2 =gm1+m2 (r1r2)n mod n2
   So, Pailler cryptosystem 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 [1].

2. Multiplicative Homomorphic Encryption

   A Homomorphic encryption is multiplicative, if: [1]

   $$\mathrm{Enc}\ (x \otimes y) = \mathrm{Enc}(x) \otimes \mathrm{Enc}(y)$$

   $$\mathrm{Enc}\ (\prod_{i=1}^{1} mi) = \prod_{i=1}^{l} \mathrm{Enc}\ (mi)$$

There are many Homomorphic encryption scheme proposed with symmetric and asymmetric encryption.Servey on that scheme listed below.

A. Goldwasser-Micali Scheme[2]

The Goldwasser–Micali (GM) cryptosystem is an asymmetric key encryption algorithm developed by Shafi Goldwasser and Silvio Micali in 1982. GM has the distinction of being the first probabilistic public-key encryption scheme which is provably secure under standard cryptographic assumptions. However, it is not an efficient cryptosystem, as ciphertexts may be several hundred times larger than the initial plaintext. To prove the security properties of the cryptosystem, Goldwasser and Micali proposed the widely-used definition of semantic security.

Goldwasser–Micali consists of three algorithms: a probabilistic key generation algorithm which produces a public and a private key, a probabilistic encryption algorithm, and a deterministic decryption algorithm.

The scheme relies on deciding whether a given value x is a square mod N, given the factorization (p, q) of N. This can be accomplished using the following procedure:

1. Compute xp = x mod p, xq = x mod q.
2. If Xp (p-1)/2=1 (mod p) and, Xq (q-1)/2=1 (mod q) then x is a quadratic residue mod N.

Key generation

The modulus used in GM encryption is generated in the same manner as in the RSA cryptosystem. (See RSA, key generation for details.)

1. Alice generates two distinct large prime numbers p and q, randomly and independently of each other.
2. Alice computes N = p q.
3. She then finds some non-residue x such that the Legendre symbols satisfy (x/p) =(x/q) =-1and hence the Jacobi symbol (x/N) is +1. The value x can for example be found by selecting random values and testing the two Legendre symbols. If (p, q) = 3 mod 4 (i.e., N is a Blum integer), then the value N − 1 is guaranteed to have the required property.

The public key consists of (x, N). The secret key is the factorization (p, q).

Message encryption

Suppose Bob wishes to send a message m to Alice:

1. Bob first encodes m as a string of bits (m1, ..., mn).
2. For every bit mi, Bob generates a random value y from the group of units modulo N, or GCD(y,N) = 1. He outputs the value Ci=y2xmi (mod N).

Bob sends the ciphertext (c1, ..., cn).

Message decryption

Alice receives (c1, ..., cn). She can recover m using the following procedure:

1. For each i, using the prime factorization (p, q), Alice determines whether the value ci is a quadratic residue; if so, mi = 0, otherwise mi = 1.

Alice outputs the message m = (m1, ..., mn).

B. Paillier Encryption Scheme [2]

The Paillier cryptosystem is invented by Pascal Paillier in 1999, is a probabilistic asymmetric algorithm for public key cryptography. The problem of computing nth residue classes is believed to be computationally difficult. The decisional composite residuosity assumption is the intractability hypothesis upon which this cryptosystem is based.

The scheme is an additive homomorphic cryptosystem; this means that, given only the public-key and the encryption of m1 and m2, one can compute the encryption of m1+ m2.

Algorithm:

The scheme works as follows:

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. This property is assured if both primes are of equivalent length, i.e., p,q Є 1|| {0,1}s-1 for security parameter .
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.

Note that the notation a/b does not denote the modular multiplication of a times the modular multiplicative inverse of b but rather the quotient of a divided by b, i.e., the largest integer value v⩾0 to satisfy the relation a>vb.

5. The public (encryption) key is .(n,g)
6. The private (decryption) key is (λ, μ)

If using p,q of equivalent length, a simpler variant of the above key generation steps would be to set g=n+1,λ=ψ(n) and , μ= ψ(n) -1 mod n where ψ(n) =(p-1)(q-1) .[1]

Encryption

1. Let be a message to be encrypted where m Є Zn
2. Select random where r Є Zn*

Compute ciphertext as: c=gm . rn mod n2

Decryption

1. Ciphertext c Є Zn2*
2. Compute message: m=L(c λ mod n2) . μ mod N

C. Damgard-Jurik Scheme[2]

The Damgård–Jurik cryptosystem is a generalization of the Paillier cryptosystem. It uses computations modulo ns+1 where n is an RSA modulus and a (positive) natural number. Paillier's scheme is the special case with s=1. The order ψ(ns+1)(Euler's totient function) of Z*ns+1 can be divided by ns. Moreover Z*ns+1 can be written as the direct product of G*H. G is cyclic and of order ns, while H is isomorphic to Z*n. For encryption, the message is transformed into the corresponding coset of the factor group G/H and the security of the scheme relies on the difficulty of distinguishing random elements in different cosets of H. Like Paillier, the security of Damgård–Jurik can be proven under the decisional composite residuosity assumption.

Key Generation

1. Choose two large prime numbers p and q randomly and independently of each other.
2. Compute n=pq and λ=lcm (p-1,q-1)..
3. Choose an element g Є Z*ns+1 such that g= (1+n) j x mod ns+1 for a known j relative prime to n and x Є H.
4. Using the Chinese Remainder Theorem, choose d such that d mod n Є Z*n and d=0 mod λ. For instance d could be λ as in Paillier's original scheme.

The public (encryption) key is (n,g).
The private (decryption) key is d.

Encryption

1. Let be a message to be encrypted where m Є Zns.
2. Select random where r Є Z*ns+1 .
3. Compute ciphertext as: c=gm. r n mod ns+1.

Decryption

1. Ciphertext c Є Z*ns+1
2. Compute cn mod ns+1.
3. Apply a recursive version of the Paillier decryption mechanism to obtain jmd. As jd is known, it is possible to compute m=(jmd) . (jd)-1 mod ns.

D. Okamoto and Uchiyama[2]

The Okamoto–Uchiyama cryptosystem was discovered in 1998 by T. Okamoto and S. Uchiyama. The system works in the group (Z/nZ)*, where n is of the form p2q and p and q are large primes.

Key generation

A public/private key pair is generated as follows:

1. Generate large primes p and q and set n=p2q.
2. Choose g Є(Z/nZ)* such that gp ≠.1 mod p2
3. Let h = gn mod n.

The public key is then (n, g, h) and the private key is the factors (p, q).

Message encryption

To encrypt a message m, where m is taken to be an element in Z/nZ

1. Select r Є Z/nZ at random. Set C=gmhr mod n

Message decryption

If we define L(x) =x-1/p, then decryption becomes m=L(cp-1 mod p2)/ L(gp-1 mod p2) mod p

REFERENCES

[1] Ms.Parin.V.Patel (Gujarat,India), Mr.Hitesh.D.Patel (Gujarat,India) "A Survey of the Homomorphic Encryption Approach for Data Security in Cloud Computing", International Journal of Engineering Development and Research (IJEDR), ISSN:2321-9939, Vol.1, Issue 1, URL :http://www.ijedr.org/papers/ IJEDR1301004.pdf

[2] Maha TEBAA, Saïd EL HAJJI and Abdellatif EL GHAZI,"Homomorphic Encryption Applied to the Cloud Computing Security", Proceedings of the World Congress on Engineering, Vol I, London, U.K. July 4 - 6, 2012

[3] http://en.wikipedia.org/wiki

[4] National Institute of Standards and Technology- Computer Security Resource Center -www.csrc.nist.gov

[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 – 2010M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[7] 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

[8] Mahadevan Gomathisankaran, Akhilesh Tyagi, Kamesh Namuduri," HORNS: A Homomorphic Encryption Scheme for Cloud Computing using Residue Number System",IEEE, 12 May 2011

[9] 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