International Journal of Engineering Research-Online
A Peer Reviewed International Journal
Articles available online http://www.ijoer.in

Vol.2., Issue.1., 2014

RESEARCH ARTICLE

ISSN:2321-7758

# ENSURING DATA SECURITY USING HOMOMORPHIC ENCRYPTION IN CLOUD COMPUTING

**K.REVANA SURESH**
M.Tech. Scholar, G. Pullaiah College of Engineering & Technology, Kurnool,
Andhra Pradesh, India.

Article Received: 01/03/2014 | Article Revised on: 16/03/2014 | Article Accepted on:17/03/2014

**K.REVANA SURESH**

**ABSTRACT**

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). It provides the full scalability, reliability, high performance and relatively low cost feasible solution as compared to dedicated infrastructures. But having many advantages for IT organizations cloud has some issues that must be consider during its deployment. The main concern is security. Cloud service provider and cloud service consumer should make sure that cloud is safe enough for all external threats so that customer does not face any problem such as loss of data or data theft. In this paper we present security issues affecting cloud computing and propose homomorphic encryption scheme an extreme approach that provides data security which is an amazing breakthrough in solving a central open problem in cryptography.

**Keywords:** Cloud computing, homomorphic encryption, data security, cryptography.

## INTRODUCTION

Cloud computing is a technology that uses internet and central remote servers to maintain data and applications. Cloud computing to put it simply, means “Internet Computing.” In cloud computing we have seen the dramatic growth of public clouds, in which the computing resources can be accessed by the general public. One of the biggest advantages of a public cloud is its virtually unlimited data storage capabilities and elastic resource provisioning. Many IT enterprises and individuals are outsourcing their databases to the cloud servers, in order to enjoy the much lower data management cost than maintaining their own data centers. It has never been easier than now that a variety of users/clients could access or share information stored in the cloud, independent of their locations. Despite enthusiasm around the cloud data service outsourcing model, its promises cannot be fulfilled until we address the serious security and privacy concerns that data owners have.

Encryption has traditionally been viewed as a mechanism that enables secure communication, namely the problem of transmitting a message from Alice to Bob over a public channel while keeping it hidden provides a way for Alice to encrypt a message into a cipher text using Bob's public key, and for Bob to decrypt the cipher text to obtain the message using his secret key. In this view of encryption schemes, access to encrypted data is all or nothing – having the secret decryption key enables one to learn the entire message, but without the decryption key, the cipher text is completely useless. The Ability to perform computations on encrypted data without being able to "see" the data. Such ability also gives rise to a number of useful applications including the ability to privately outsource arbitrary computations to the "cloud" and the ability to store all data encrypted and perform computations on encrypted data, decrypting only when necessary. Homomorphic encryption is a special type of encryption system that permits arbitrarily complex computation on encrypted data.

## 2. ASSUMPTONS & HYPOTHESIS

**A. Problem:** Sensitive data in a cloud computing environment emerge as major issues with regard to security in a cloud based system. Firstly, whenever a data is on a cloud, anyone from anywhere anytime can access data from the cloud. Since data may be common, private and sensitive data in a cloud. So at the same time, many cloud computing service consumer and provider accesses and modify data. Thus there is a need of some data integrity method in cloud computing.

**Solution:** Encryption is a promising way to protect the confidentiality of the outsourced data, but it is much difficulty to performing effective searches over encrypted information, with complex query conditions, and care needs to be taken when using them because of the potential privacy leakages about the data owners to the data users or the cloud server. In homomorphic based encryption technique as computations are performed on cipher text resulting in confidentiality of data.

**B. Problem:** Data stealing is a one of serious issue in a cloud computing environment. Many cloud service provider do not provide their own server instead they acquire server from other service providers due to it is cost affective and flexible for operation and cloud provider. So there is a much probability of data can be stolen from the external server.

**Solution:** Our implementation in this technique is encrypting plain text by means of a key so thus providing security for cloud stored data.

## 3. RESEARCH & METHODOLOGY

### A.LITERATURE SURVEY

Cloud computing to put it simply, means "Internet Computing." The Internet is commonly visualized as clouds; hence the term "cloud computing" for computation done through the Internet. With Cloud Computing users can access database resources via the Internet from anywhere, for as long as they need, without worrying about any maintenance or management of actual resources. Besides, databases in cloud are very dynamic and scalable. Cloud computing provides the facility to access shared resources and common infrastructure, offering services on demand over the network The location of physical resources and devices being accessed are typically not known to the end user. It also provides facilities for users to develop, deploy and manage their applications 'on the cloud', which entails virtualization of resources that maintains and manages itself.

Cloud computing can be visualized as a pyramid consisting of 3 sections:

1. cloud Application
2. cloud platform
3. cloud Infrastructure

**1. Cloud Application**

This is the apex of the cloud pyramid, where applications are run and interacted with via a web browser, hosted desktop or remote client. A hallmark of commercial cloud computing applications is that users never need to purchase expensive software licenses themselves. Instead, the cost is incorporated into the subscription fee. A cloud application eliminates the need to install and run the application on the customer's own computer, thus removing the burden of software maintenance, ongoing operation and support.

**2. Cloud Platform**

The middle layer of the cloud pyramid, which provides a computing platform or framework as a service. A cloud computing platform dynamically provisions, configures, reconfigures and de-provisions servers as needed to cope with increases or decreases in demand. This in reality is a distributed computing model, where many services pull together to deliver an application or infrastructure request.

**3. Cloud Infrastructure**

The foundation of the cloud pyramid is the delivery of IT infrastructure through virtualization. Virtualization allows the splitting of a single physical piece of hardware into independent, self governed environments, which can be scaled in terms of CPU, RAM, Disk and other elements. The infrastructure includes servers, networks and other hardware appliances delivered as either infrastructure, ”web services”, ”farms” or “cloud centers”. These are then interlinked with others for resilience and additional capacity.

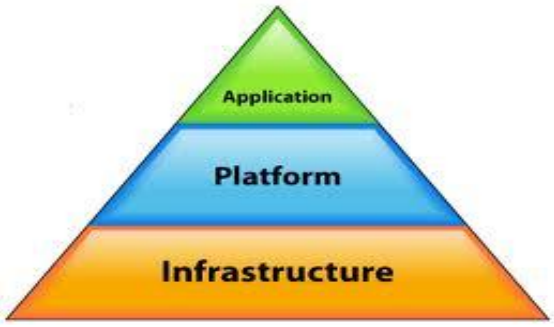



Fig.1: Pyramid of Cloud Computing

**B. Services and Deployment:** In cloud computing, everything is delivered *as a* Service from testing and security, to collaboration and Meta modeling. The cloud was rapidly becoming a conflagration of buzzwords “as a service”. Today there are three main service models

**(i) Software as a Service (SaaS).** SaaS is a software model provided by the vendor through an online service. User doesn’t have to install or maintain SaaS application. Software is running on a provider’s cloud infrastructure and a user can access it via web browser.

**(ii) Platform as a Service (PaaS).** PaaS enables companies to develop applications more quickly and efficiently in a cloud environment using programming languages and tools supported by the provider.

**(iii) Infrastructure as a Service (IaaS).** With IaaS, a company can rent fundamental computing resources for deploying and running applications or storing data. It enables companies to deliver applications more efficiently by removing the complexities involved with managing their own infrastructure.

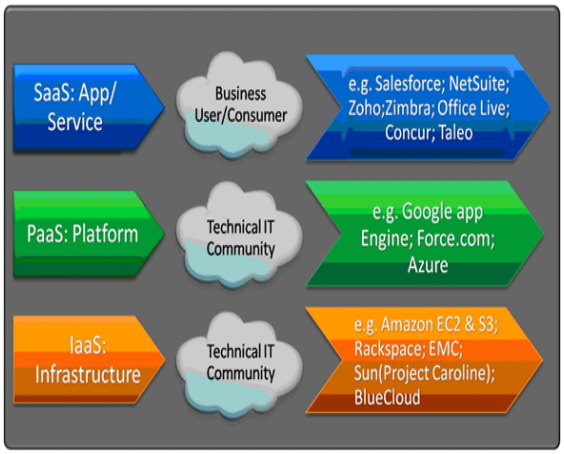



**Fig. 2: Architecture of Cloud Service models**

**C. Deployment models:**

Each company chooses a deployment model for a cloud computing solution based on their specific business, operational, and technical requirements. There are four primary cloud deployment models: private cloud, community cloud, public cloud, and hybrid cloud. Here is how each of the deployment models is defined:

**(a)Private cloud:** Private cloud (also referred to as 'corporate' or 'internal' cloud) is a term used to denote a proprietary computing architecture providing hosted services on private networks. This type of cloud computing is generally used by large companies, and allows their corporate network and data centre administrators to effectively become in-house 'service providers' catering to 'customers' within the corporation.

**(b) Community cloud:** A community cloud refers to cloud computing environment shared and managed by several organizations that have similar requirements and are sharing the infrastructure in order to realize some of the benefits of cloud computing. With the costs spread over fewer users than a public cloud this option is more expensive but may offer a higher level of privacy, security and/or policy compliance. It may be managed by the company or a third party and can exist on or off premise.

**(c) Public cloud:** Public cloud *or* multi-tenant cloud describes cloud computing in the traditional mainstream sense. Resources are dynamically provisioned on a fine-grained, self-service basis over the Internet, via web services, from an off-site third-party provider who shares resources and bills on a fine-grained utility computing basis. The cloud infrastructure is owned by a cloud vendor, and is accessible to the general public or a large industry group.

**(d) Hybrid cloud:** A hybrid cloud environment consists of multiple clouds (private, community, or public) and is the typical cloud deployment model for most enterprises. By integrating multiple cloud services users may be able to ease the transition to public cloud services while avoiding issues such as PCI compliance. The main benefit of the hybrid cloud is that it provides the scalability and low costs of a public cloud without exposing mission-critical applications and data to third-party.

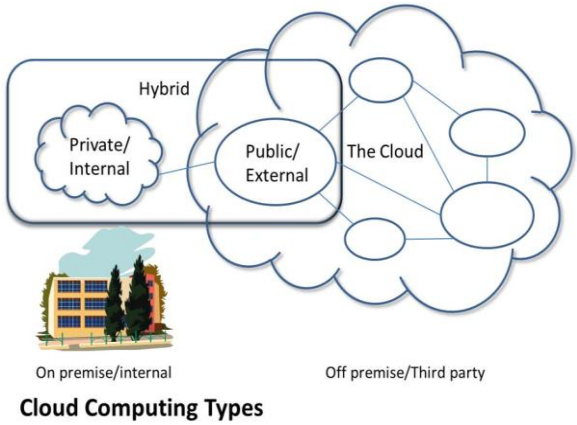



**Fig. 3: Cloud Computing Types**

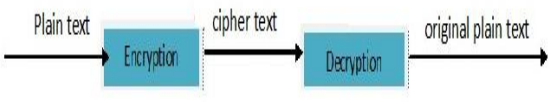
**4. ENCRYPTION AND DECRYPTION**

In cryptography, encryption is the process of encoding messages (or information) in such a way that eavesdroppers or hackers cannot read it, but that authorized parties can. In an encryption scheme, the message or information (referred to as plaintext) is encrypted using an encryption algorithm, turning it into an unreadable cipher text .This is usually done with the use of an encryption key, which specifies how the message is to be encoded. Any adversary that can see the cipher text should not be able to determine anything about the original message. An authorized party, however, is able to decode the cipher text using a decryption algorithm that usually requires a secret decryption key that adversaries do not have access to. For technical reasons, an encryption scheme usually needs a key-generation algorithm to randomly produce keys.

There are two basic types of encryption schemes: Symmetric-key and public-key encryption. In symmetric-key schemes, the encryption and decryption keys are the same. Thus communicating parties must agree on a secret key before they wish to communicate. In public-key schemes, the encryption key is published for anyone to use and encrypt messages. However, only the receiving party has access to the decryption key and is capable of reading the encrypted messages. Public-key encryption is a relatively recent invention: historically, all encryption schemes have been symmetric-key (also called private-key) schemes.

Encryption, by itself, can protect the confidentiality of messages, but other techniques are still needed to protect the integrity and authenticity of a message; for example, verification of a message authentication codes (MAC) or a digital signature. Standards and cryptographic software and hardware to perform encryption are widely available, but successfully using encryption to ensure security may be a challenging problem.

Encryption and decryption are both methods used to ensure the secure passing of messages and other sensitive documents and information. Encryption basically means to convert the message into code or scrambled form, so that anybody who does not have the 'key' to unscramble the code, cannot view it. This is usually done by using a 'cipher'. A cipher is a type of algorithm used in encryption that uses a certain described method to scramble the data. The cipher can only be 'deciphered' with a 'key'. A key is the actual 'described method' that was used to scramble the data, and hence the key can also unscramble the data. When the data is unscrambled by the use of a key, that is what is known as 'decryption'. It is the opposite of encryption and the 'described method' of scrambling is basically applied in reverse, so as to unscramble it. Hence, the jumbled and unreadable text becomes readable once again.



Encryption with key
Encryption key: $K_E$
Decryption key: $K_D$
$C = E(K_E, P)$
$P = D(K_D, E(K_E, P))$

**4.1 HOMOMORPHIC ENCRYPTION**

Homomorphic Encryption allows access to highly scalable, inexpensive, on-demand computing resources that can execute the code and store the data that are provided to them. This aspect, known as data outsourced computation is very attractive, as it alleviates most of the burden on IT services from the consumer. Nevertheless, the adoption of data outsourced computation by business has a major obstacle, since the data owner does not want to allow the un trusted cloud provider to have access to the data being outsourced. Merely encrypting the data prior to storing it on the cloud is not a viable solution, since encrypted data cannot be further manipulated. This means that if the data owner would like to search for particular information, then the data would need to be retrieved and decrypted a very costly operation, which limits the usability of the cloud to merely be used as a data storage centre.

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.

Definition: An encryption is homomorphic, if: from Enc(a) and Enc(b) it is possible to compute Enc(f (a, b)), where f can be: +, ×, ⊕ and without using the private key.

For plaintexts P1 and P2 and corresponding ciphertext C1 and C2, a homomorphic encryption scheme permits meaningful computation of P1 Θ P2 from C1 and C2 without revealing P1 or P2.The cryptosystem is additive or multiplicative homomorphic depending upon the operation Θ which can be addition or multiplication**.**

A homomorphic encryption scheme consists of the following four algorithms:

**KeyGen** (λ):

- Input-the security parameter λ.
- Output-a tuple (sk, pk) consisting of the secret key sk and public key pk .

**Encrypt** ( pk,π ):

- Input-a public key pk and a plaintext π .
- Output-ciphertext Ψ.

**Decrypt** (sk, Ψ):

- Input-a secret key sk and a ciphertext Ψ.
- Output-the corresponding plaintext π .

**Evaluate** (pk ,C ,Ψ ):

- Input-a public key , a circuit with inputs and a set Ψ of ciphertext ,Ψ1 . . . . . Ψt

Output-a ciphertext Ψ.

Therefore, a homomorphic encryption scheme consists of all algorithms of a conventional public key encryption scheme and an extra one. The correctness-condition for the conventional part of a homomorphic encryption scheme is identical to that of a (non-homomorphic) public key encryption scheme.

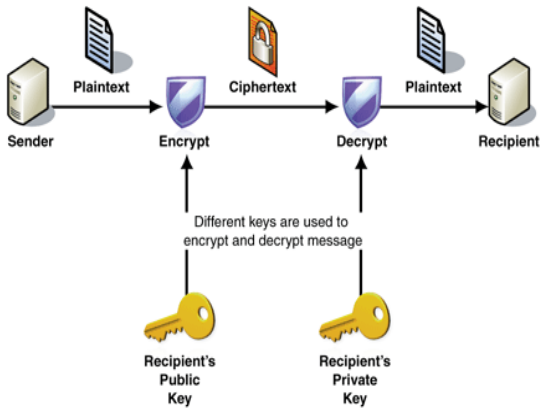



**Fig. 5: Working of Homomorphic Encryption Algorithm**

**4.2 Additive homomorphic encryption**

Additive homomorphic encryption is implemented by Paillier Cryptosystem. The Paillier cryptosystem, named after and invented by Pascal Paillier in 1999, is a probabilistic asymmetric algorithm for public key cryptography. The problem of computing n-th 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 $\epsilon$ 1||{0,1}$^{s-1}$for security parameter$S$
2. Compute n=pq and λ=lcm(p-1,q-1)
3. Select random integer g where g $\epsilon Z_{n2}^*$
4. Ensure n divides the order of g by checking the existence of the following modular multiplicative inverse,$\mu=(L(g^{\lambda} \bmod n^2))^{-1} \bmod n$
   where function Lis 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 bbut rather the quotient of a divided by b , i.e., the largest integer value v>=0 to satisfy the relation a>=vb.

If using p,q of equivalent length, a simpler variant of the above key generation steps would be to set g=n+1, λ =ϕ(n)and $\mu= \phi(n)^{-1}$, where ϕ(n)=(p-1)(q-1).

**Encryption**

1. Let m be a message to be encrypted where m $\epsilon Z_n$
2. Select random $r$where $r \epsilon Z_n^*$
3. Compute ciphertext as: $c=g^m.r^n \bmod n^2$

**Decryption**

1. Ciphertext c$\epsilon$ $Z_{n2}^*$
2. Compute message: $m=L(c^{\lambda} \bmod n^2).\ \mu \bmod n$

Application of additive homomorphic encryption is electronic voting.

**4.3 Multiplicative Homomorphic Encryption**

Additive homomorphic encryption is implemented by RSA cryptosystem

RSA is an algorithm for public-key cryptography that is based on the presumed difficulty of factoring large integers, the factoring problem. RSA stands for Ron Rivest, Adi Shamir and Leonard Adleman, who first publicly described the algorithm in 1977. Clifford Cocks, an English mathematician, had developed an equivalent system in 1973, but it was classified until 1997.A user of RSA creates and then publishes the product of two large prime numbers, along with an auxiliary value, as their public key. The prime factors must be kept secret. Anyone can use the public key to encrypt a message, but with currently published methods, if the public key is large enough, only someone with knowledge of the prime factors can feasibly decode the message. Whether breaking RSA encryption is as hard as factoring is an open question known as the RSA problem.

**Key generation**

RSA involves a public key and a private key**.** The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted in a reasonable amount of time using the private key. The keys for the RSA algorithm are generated the following way:

1. Choose two distinct prime numbers p and q.
   - For security purposes, the integers p and q should be chosen at random, and should be of similar bit-length. Prime integers can be efficiently found using a primality test.
2. Compute n = pq.
   - n is used as the modulus for both the public and private keys. Its length, usually expressed in bits, is the key length.
3. Choose an integer e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1$; i.e., e and $\phi(n)$ are coprime.
   - e is released as the public key exponent.
   - e having a short bit-length and small Hamming weight results in more efficient encryption – most commonly $2^{16} + 1 = 65,537$. However, much smaller values of e (such as 3) have been shown to be less secure in some settings.
4. Determine d as $d \equiv e^{-1} \pmod{\phi(n)}$, i.e., d is the multiplicative inverse of e (modulo $\phi(n)$)

**Encryption**

Alice transmits her public key (n, e) to Bob and keeps the private key secret. Bob then wishes to send message M to Alice. He first turns M into an integer m, such that $0 \le m < n$ by using an agreed-upon reversible protocol known as a padding scheme. He then computes the ciphertext c corresponding to

$$C=m^e \pmod{n}$$

This can be done quickly using the method of exponentiation by squaring. Bob then transmits *c* to Alice.

**Decryption**

Alice can recover *m* from *c* by using her private key exponent *d* via computing

$$M=c^d \pmod{n}$$

**5. CONCLUSION**

The cloud computing security based on 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. Some important security services including authentication, encryption and decryption are provided in Cloud Computing system.

**REFERENCES**

[1] Vic (J.R.) Winkler, "Securing the Cloud, Cloud Computer Security, Techniques and Tactics", Elsevier, 2011.

[2] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In 18th Annual Eurocrypt Conference (EUROCRYPT'99), Prague, Czech Republic, volume 1592, 1999

[3] Julien Bringe and al. An Application of the Goldwasser-Micali Cryptosystem to Biometric Authentication, Springer-Verlag, 2007.

[4] R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. Communications of the ACM, 21(2):120-126, 1978. Computer Science, pages 223-238. Springer, 1999.

[5] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, 469-472, 1985.

[6] Craig Gentry, A Fully Homomorphic Encryption Scheme, 2009.

[7] Rivest R., Adleman L. and Dertouzos M., (1978), "On data banks and privacy homomorphisms"Foundations of Secure Computation, Academic Press.

[8] Brickell, E. and Yacobi, Y., (1987). "On privacy homomorphisms", Advances in Cryptology volume 304 of Lecture Notes in Computer Science, Springer, New York, USA,

[9] Rappe, D., (2004), Homomorphic Cryptosystems and their Applications, Ph.D. thesis, University of Dortmund, Dortmund, Germany.