IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.7, Special Issue.2, Jan 2019 E-ISSN: 2347-2693

Cloud Security System With Sequel Homomorphic Encryption and Diffie-Hellman Algorithm

K Kuppuswamy[1*], M.R. Nagarajan[2]

[1,2]Department of Computer Science, Alagappa University, Karaikudi, India

[*]*Corresponding Author: kkdiksamy@yahoo.com, Tel.:* + 91-*04565-223100-662*

Available online at: www.ijcseonline.org

***Abstract*— A set of resources and services offered over the Internet, is called cloud computing. These computing services are delivered from data centers located across the world. Cloud computing consumers benefited by providing virtual resources via internet such as Platform, Infrastructure and Software as a Service. The consequential challenge in cloud computing are the privacy and security issues caused by its multi-tenancy nature and the outsourcing of infrastructure, sensitive data. It is available on Pay-Per-Use model. Various malicious activities from illegal users have threatened this technology such as data misuse, inflexible access control mechanism. The occurrence of these threats may result into spoil or illegal access of critical and private data of end user and business user. In this paper, we identify the most vulnerable security issues or attribute in cloud computing architecture, which will enable both end users and vendors to know about the key security threats associated with cloud computing and propose relevant solution directives to strengthen security in the cloud computing environment. We propose secure cloud architecture by using sequel homomorphic encryption scheme with diffie-hellman algorithm for organizations to strengthen the security mechanism**.

***Keywords*—** *Secure Cloud Computing Architecture, Security, Sequel Homomorphic Encryption, Diffie-Hellman algorithm*

I. Introduction

Information Security achieved using Fully Homomorphic Encryption. The cloud service provider and client prefers this encryption method. Homomorphic encryption - an operation performed on a set of cipher texts such that decrypting the result of the operation is the same as the result of some operation performed on the plaintexts [1]. The decisional Diffie–Hellman (DDH) assumption is a computational hardness assumption about a certain problem involving discrete logarithms in cyclic groups. It is used as the basis to prove the security of many cryptographic protocols, most notably the ElGamal and Cramer–Shoup cryptosystems. The problem statement: Given g^x and g^y for uniformly and independently chosen x, y random integers. The value of g^{xy} “looks like” a random element in G. where g is a generator and G is a multiplicative cyclic group [2]. Since Cloud is maintained by third parties, security and privacy plays a vital role in making cloud a popular and successful technology. A sequel homomorphic encryption can be used in order to achieve rich security and with minimized response time.

II. Related Work

Cloud computing is an information technology (IT) paradigm, a model for enabling ubiquitous access to shared pools of configurable resources (such as computer networks, servers, storage, applications and services), which can be rapidly provisioned with minimal management effort, often over the Internet. Based on a service model that the cloud is offering IaaS (Infrastructure-as-a-Service), PaaS (Platform-as-a-service), SaaS (Software-as-a-Service),etc. Cloud can be deployed with the model of Public, Private and Hybrid.Since all the data is transferred using Internet, data security is of major concern in the cloud. Here are key mechanisms for protecting data via Access Control, Auditing, Authentication, and Authorization (AAAA). If data is secure, should fulfil three conditions (i) Confidentiality (ii) Integrity (iii) Availability.

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. Each homomorphic encryption algorithm is developed mainly based on two factors: Hardness of breaking the security of encryption and Efficiency (in terms of time) of execution of encryption, decryption processes. To secure the encryption or decryption process, different mathematical constructs are used like composite residuocity, high degree residuocity and quadratic residuocity. Trapdoor functions are used to secure data in homomorphic encryption. They are one way functions used during encryption and decryption process. Different mathematical constructs are used to create trapdoors.Efficiency of homomorphic encryption schemes is considered to be good if generated cipher text size is small and time taken to run encryption, decryption or recryption process is less.

III. Cloud Computing And Security

Cloud computing is the delivery of computing services over the Internet. Cloud services allow individuals and businesses to use software and hardware that are managed by third parties at remote locations. The cloud computing model allows access to information and computer resources from anywhere that a network connection is available. Cloud computing provides a shared pool of resources, including data storage space, networks, computer processing power, and specialized corporate and user applications.

Examples for cloud services such as online file storage, social networking sites, webmail, and online business applications. The cloud computing model allows access to information and computer resources from anywhere that a network connection is available. Cloud computing provides a shared pool of resources, including data storage space, networks, computer processing power, and specialized corporate and user applications[3],[4] .

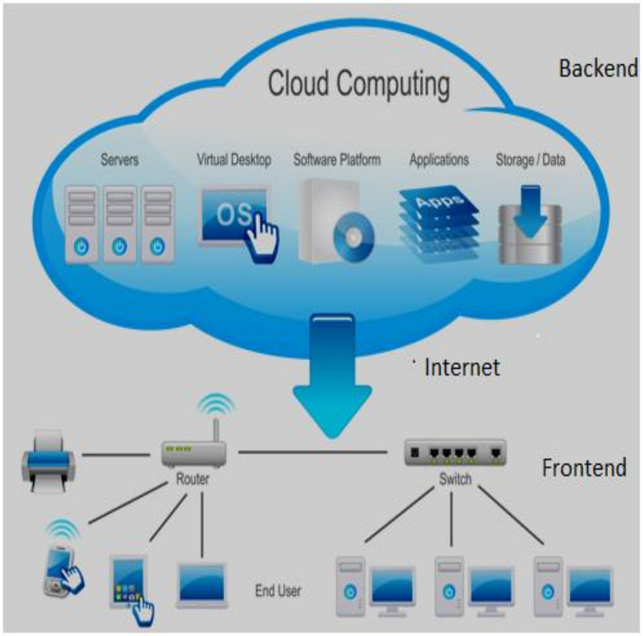

Figure 1. Architecture of Cloud Computing

A. Types of Cloud

Public cloud is offered over the Internet and are owned and operated by a cloud provider. Some examples include services aimed at the general public, such as online photo storage services, e-mail services, or social networking sites.

Private cloud, the cloud infrastructure is operated solely for a specific organization, and is managed by the organization or a third party.

Community cloud, the service is shared by several organizations and made available only to those groups. The infrastructure may be owned and operated by the organizations or by a cloud service provider.

Hybrid cloud is a combination of different methods of resource pooling (for example, combining public and community clouds).

B. Security Challenges

Today, cloud computing facing many security challenges including data segregation, authentication, and data recovery, data management. Public cloud increases the highest data explosion and it should be managed properly. Security is main issue because the device used to serve the service doesn't belong to the user themselves. This is great concern when user have valuable and personal information stored in cloud computing services.

Data Security:

Data security is a significant task with a lot of complexity. Data protection method such as redaction, truncations, obfuscation, and others, should be taken with great concern. Homomorphism encryption can be used for data security. But with this key management is a problem [5].

Data Segregation:

Encrypted data from multiple companies or user may be stored on same hard disk so a mechanism should be there to protect data.

Data Recover:

Every service provider should have the data recovery mechanism to recover user data from any disaster [6].

Data integration:

Data integrity comprises the following cases, when some human errors occurs when data is entered. Errors may occur when data is transmitted from one computer to another; otherwise error can occur from some hardware malfunctions, such as disk crashes.Software bugs or viruses can also make viruses. 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 [7].

Authentication:

As user keep their valuable information on cloud service across the internet. It can be accessible by unauthorized people. Hence authenticating user cloud should have identity management system [8].

IV. Diffie-Hellman Algorithm

This algorithm uses arithmetic modulus as the basis of its calculation. Suppose Alice and Bob follow this key exchange procedure with Eve acting as a man in middle interceptor (or the bad guy).

Here are the calculation steps followed in this algorithm that make sure that eve never gets to know the final keys through which actual encryption of data takes place.

- First, both Alice and Bob agree upon a prime number and another number that has no factor in common. Lets call the prime number as **p** and the other number as **g.** Note that **g** is also known as the generator and **p** is known as prime modulus.
- Now, since eve is sitting in between and listening to this communication so eve also gets to know **p** and **g**.
- Now, the modulus arithmetic says that **r** = (**g** to the power **x**) mod **p.** So **r** will always produce an integer between 0 and **p**.
- The first trick here is that given **x** (with **g** and **p** known) , its very easy to find **r**. But given **r** (with **g** and **p** known) its difficult to deduce **x**.
- One may argue that this is not that difficult to crack but what if the value of **p**is a very huge prime number? Well, if this is the case then deducing **x** (if **r** is given) becomes almost next to impossible as it would take thousands of years to crack this even with supercomputers.
- This is also called the discrete logarithmic problem.
- Coming back to the communication, all the three Bob, Alice and eve now know **g** and **p**.
- Now, Alice selects a random private number **xa** and calculates (**g** to the power **xa**) mod **p** = **ra**. This resultant **ra** is sent on the communication channel to Bob. Intercepting in between, eve also comes to know **ra**.
- Similarly Bob selects his own random private number **xb**, calculates (**g** to the power **xb**) mod **p** = **rb** and sends this **rb** to Alice through the same communication channel. Obviously eve also comes to know about **rb**.
- So eve now has information about **g**, **p**, **ra** and **rb**.
- Now comes the heart of this algorithm. Alice calculates (**rb** to the power **xa**) mod **p** = **Final key** which is equivalent to **(g to the power (xa*xb)) mod p .**
- Similarly Bob calculates **(ra to the power xb) mod p** = **Final key** which is again equivalent to **(g to the power(xb * xa)) mod p**.
- So both Alice and Bob were able to calculate a common **Final key** without sharing each others private random number and eve sitting in between will not be able to determine the **Final key** as the private numbers were never transferred.

As explained above the Diffie-Hellman algorithm works perfectly to generate cryptographic keys which are used to encrypt the data being communicated over a public channel. The simplest and the original implementation[2] of the protocol uses the multiplicative group of integers modulo *p*, where *p* is prime, and *g* is a primitive root modulo *p*. These two values are chosen in this way to ensure that the resulting shared secret can take on any value from 1 to *p*–1. Here is an example of the protocol, with non-secret values in blue, and secret values in **red**.

1. Alice and Bob agree to use a modulus $p = 23$ and base $g = 5$ (which is a primitive root modulo 23).
2. Alice chooses a secret integer $\boldsymbol{a} = 4$, then sends Bob $A = g^{a} \bmod p$
   - $A = 5^{4} \bmod 23 = 4$
3. Bob chooses a secret integer $\boldsymbol{b} = 3$, then sends Alice $B = g^{b} \bmod p$
   - $B = 5^{3} \bmod 23 = 10$
4. Alice computes $\boldsymbol{s} = B^{a} \bmod p$
   - $\boldsymbol{s} = 10^{4} \bmod 23 = 18$
5. Bob computes $\boldsymbol{s} = A^{b} \bmod p$
   - $\boldsymbol{s} = 4^{3} \bmod 23 = 18$
6. Alice and Bob now share a secret (the number 18).

Both Alice and Bob have arrived at the same value s, because, under mod p,

More specifically, Note that only *a*, *b*, and ($g^{ab} \bmod p = g^{ba} \bmod p$) are kept secret. All the other values – *p*, *g*, $g^{a} \bmod p$, and $g^{b} \bmod p$ – are sent in the clear. Once Alice and Bob compute the shared secret they can use it as an encryption key, known only to them, for sending messages across the same open communications channel. Of course, much larger values of *a*, *b*, and *p* would be needed to make this example secure, since there are only 23 possible results of *n* mod 23. However, if *p* is a prime of at least 600 digits, then even the fastest modern computers cannot find *a* given only *g*, *p* and $g^{a} \bmod p$. Such a problem is called the discrete logarithm problem. The computation of g^{a}mod *p* is known as modular exponentiation and can be done efficiently even for large numbers. Note that *g* need not be large at all, and in practice is usually a small integer (like 2, 3, ...).

V. METHODOLOGY

In a simple mathematics, homomorphic encryption can be denoted as

- **Homomorphic Encryption**
  - encryption function ε
  - plaintext xn
  - cipher text cn such that ε(xn) = cn
- **Homomorphic Addition**

  A given cryptosystem is considered additively homomorphic

  iff ∃∆: ε(x1) ∆ ε(x2) = ε(x1 + x2)
- **Homomorphic Multiplication**

  A given cryptosystem is considered multiplicatively homomorphic

  iff ∃∆: ε(x1) ∆ ε(x2) = ε(x1 x2)

- **Modified or Sequel Homomorphic Multiplication**
  A given cryptosystem is considered multiplicatively homomorphic
  iff ∃∆: [ε(x1) x n_i]∆ [ε(x2) x n_i]= [ε(x1 x2) x n_i]
  Where n_i **:** 1,2,3…n and i refers Successive codes to be encrypted

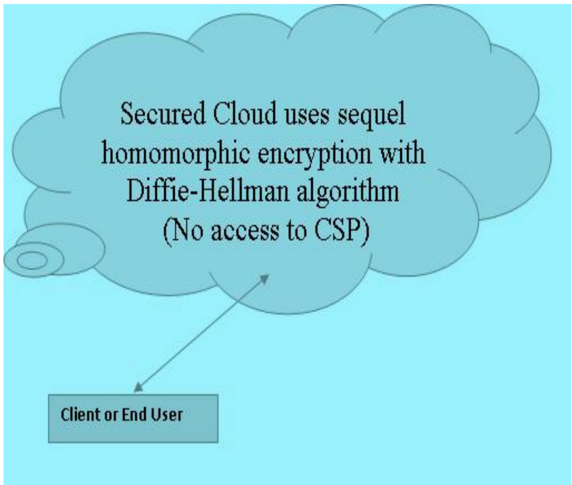

Figure 2. Secured Cloud Architecture uses sequel homomorphic encryption with Diffie Hellman Algorithm

This study is focused on to develop a model for sequel fully homomorphic disk encryption schemes. Which help in enhanced privacy information retrieval, queries to search engine, spam filtering with secured cloud service provider. The new model provides efficient key storage and key management with the help of Diffie- Hellman algorithm.

VI. Results and Discussion

Only authorized users can encrypt or decrypt the data. For encryption and decryption of data, basic sequel homomorphic encryption scheme is used along with Diffie-Hellman algorithm. The Diffie-Hellman algorithm is used for secure channel establishment and for mutual authentication. According to this thesis a secure channel should be established two users to transfer the data. To implement this architecture of cloud computing MATLAB is used.

VII. Conclusion and Future Scope

An anonymous authentication and authorization protocol is designed using anonymous public key certificates along with standard Strong Authentication and XACML servers. The same encryption scheme may be applied to the SQL queries for cloud database in future work or the load of VM can be reduced by adding any other factor without compromising the security of the data.

References

[1] Aditi Soral, "*Achieving Fully Homomorphic Encryption in Security -A Survey*", SSRG International Journal of Computer Science and Engineering (SSRG-IJCSE) , Vol.**3,** Issue.**2**, pp. **2348 – 8387**.

[1] Anjana Chaudhary, Ravinder Thakur, Manish Mann "*Security In Cloud Computing By Using Sequel Homomorphic Encryption Scheme With Diffie-Hellman Algorithm*", International Journal of Advanced Computational Engineering and Networking, Vol.**2,** Issue.**10**, pp. **2320-2106**.

[2] Umer Khalida,Abdul Ghafoor, Misbah Irum, Muhammad Awais Shibli "*Cloud based Secure and Privacy Enhanced Authentication & Authorization Protocol*", ScienceDirect, Procedia Computer Scurity, **pp** 22(2103) 680 – 688

[3] Kashif Munir1, Dr. Sellapan Palaniappan, "*Secure Cloud Architecture*", International Journal of Advanced Computing Vol.**4,** Issue.**1.**

[4] Tim Mather, Subra Kumaraswamy, Shahed Latif Cloud Security and Privacy: An Enterprise perspective of Risks and Compliance, O'Reilly Media, Inc.,2009.

[5] L.Arockiam,S.Monikandan & G.Parthasarathy "Cloud Computing: A Survey" http://interscience.in/IJIC_Vol1Iss2/paper5.pdf

[6] Anitha Y "Security Issues in Cloud Computing - A Review" in International Journal of Thesis Projects and Dissertations (IJTPD) Vol. 1, Issue 1, PP: (1-6),Month: October-December 2013.

[7] R. Kalaichelvi Chandrahasan, S Shanmuga Priya and Dr. L. Arockiam "Research Challenges and Security Issues in Cloud Computing" in International Journal of Computational Intelligence and Information Security, March 2012 Vol. 3, No. 342

Authors Profile

Dr. K, Kuppusamy working as pforessor in Department of Computer Science at Alagappa University, Karaikudi – 630 003. Tamil Nadu, India.. He is a member of Professional bodies , Edutorail board and Reviewer Society. He has published more than 20 research papers in reputed international journals including IJARTET, IJCSET,etc and conferences including IEEE and it's also available online. His main research work focuses on Network / Infomration Security, Neural Networks, and Operation Research. He has 29 years of teaching experience and 15 years of Research Experience.

Mr Nagarajan M R working as Consultant in Virtusa Consulting Services Pvt Ltd, Chennai , Tamil Nadu, India. His main research work focuses on Network / Infomration Security, Neural Networks, CI,Big Data, Machine Learning and Operation Research. He has 5 years of teaching experience and 2 years of Research Experience across diverse technology.