ENSURING SECURITY PROOF IN CLOUD NETWORK

R.Manjusha

Research Scholar, Department of Information Technology,
Sathyabama University,Chennai,Tamil Nadu,India
Manjusha84rr@yahoo.co.in

R.Ramachandran

Professor (ECE), &Director (Research),,
Sri Venkateshwara college of Engineering, Chennai, Tamilndu. India
chandra557@yahoo.co.in

Abstract

Cloud Computing is developed and used by all enterprise in world to outsource their data. To maintain confidentiality of shared data against entrusted party encryption of data is important. Many scheme is introduced to achieve security of outsource data in cloud .But they fails to achieve data integrity in cloud. Sometimes entrusted cloud service provider modifies or deletes the data without getting permission from owner in cloud. In this paper we investigate the basic problem of data integrity in cloud. Homomorphic Hierarchical Attribute Based Encryption with integrity proof is proposed in this paper to avoid security risk in cloud. Performance of Homomorphic Hierarchical Attribute Based Encryption is compared with key-aggregate cryptosystem (KAC) and Hierarchical Attribute based encryption(HABE) scheme in cloudsim. Based on data integrity, confidentiality and Validity of data best encryption algorithm is chosen in cloud

Keywords: Cloud computing, Homomorphic Encryption, Hierarchical Attribute Based Encryption, Cloudsim

1. Introduction

Cloud Computing is emerging computing technology in Information technology industry. Cloud computing provides three types of services namely software as services, platform as services and infrastructure as services. The example of software as services is Salesfore, Google Apps, Microsoft Office 365, infrastructure as service is Amazon EC2, Windows Azure, Rack space, Google Compute Engine, and platform as service is services Google apps engine and Windows azure. There are four types of cloud they are private cloud, public cloud, community cloud and hybrid cloud. Private cloud is owned by enterprise or it is leased by enterprise that is private organization owns the infrastructure. Public cloud access by any one in public networks in mega scale infrastructure[1] .Community cloud is shared by many organizations or enterprise which shares the resources that is managed by third party. In cloud user can store and share data[18]. Security of data is important in cloud; to secure data in this paper Homomorphic Hierarchical Attribute Based Encryption with integrity proof is proposed. In section 2 related work of Homomorphic Hierarchical Attribute Based Encryption with integrity proof is explained. In section 3 architecture of Homomorphic Hierarchical Attribute Based Encryption with integrity proof is explained. In section 4 implementation of Homomorphic Hierarchical Attribute Based Encryption with integrity proof is explained. In section 5 using performance evolution of Homomorphic Hierarchical Attribute Based Encryption is compared with key policy Attribute Based Encryption, cipher text policy attributes based encryption and in section 6 conclusion of this paper explained

2. Related work

Secure data sharing in cloud using homomorphic hierarchical attribute based encryption with integrity proof The survey on attribute based encryption techniques is presented in this section. In key policy attribute based encryption techniques private key is associated with policy and cipher text is associated with attribute. Decryption of cipher text is possible if and only if attribute satisfy policy[2]. Fine grained access control is achieved by key policy attribute based encryption but cannot able to achieve flexibility and scalability. In cipher text policy attribute based encryption cipher text is associated with access policy and private key is associated with attributes. Decryption is possible only when attribute satisfies the policy[13]. Flexibility and scalability is needed to improve in cipher text policy attribute based encryption. Hierarchical attribute based encryption is combination of identity based encryption and cipher text policy attribute based encryption. Hierarchical attribute based encryption achieves fine grained access control, flexibility and scalability but does not have integrity proof [3]. Homomorphic encryption is strongest encryption techniques in cloud. In homomorphic encryption plaintext are converted into cipher text and it worked in original form. Homomorphic encryption is strongest encryption techniques in cloud [1].

3. Architecture of homomorphic hierarchical attribute based encryption with integrity proof

The network architecture of homomorphic attribute based encryption has four different entities they are data owner, cloud service provider, third party auditor and users[4]. Data owner using homomorphic hierarchical attribute based encryption encrypts the data and stores in cloud service provider. The Cloud service provider provides computation resources, services and storage space f or both data owner and users. Third party auditor has more access capabilities than users and trusted by data owner. User can access the data and decrypts the data using homomorphic hierarchical attribute based encryption and verify the data received is send by data owner [12].

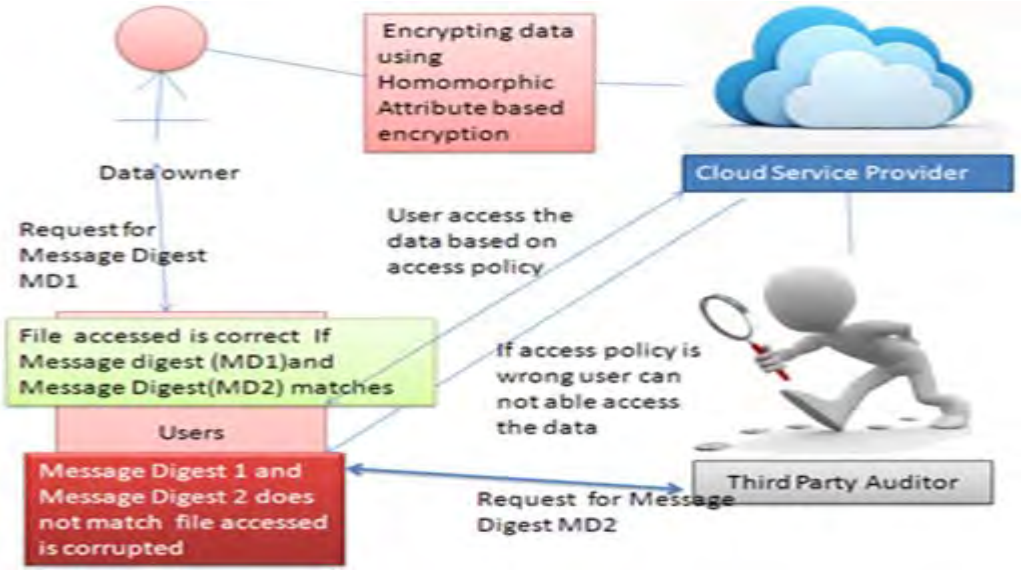

Fig. 1. Architecture Of Homomorphic Hierarchical Attribute Based Encryption With Integrity Proof

In cloud service provider data owner encrypts data using homomorphic hierarchical attribute based encryption that is running in distributed, cooperated and simultaneous manner. In homomorphism encryption complex mathematical operation are allowed without compromising the encryption. The access of file is based on homomorphic hierarchical attribute based encryption. If user's access policy matches and satisfies with hierarchical attributes then access to the file is given to users[5]. For more security in this paper third party auditor is introduce and user request message digest for that file, user has to access from both third party auditor and data owner[11]. Message digests with read access received by third party auditor and Message digest with read access received by data owner are same then file accessed by user is correct and no alteration is done in the file. Message digests with read access received by third party auditor and Message digest with read access received by data owner are not same then file accessed by user is corrupted and some alteration is done in the original file.

4. Implementation Of Homomorphic Hierarchical Attribute Based Encryption With Integrity Proof

In this section we are going to discuss some main operation like encrypting file using homomorphic attribute based encryption, From user getting message digest for file, Accessing file using homomorphic attribute based encryption and Checking integrity of the file.

4.1. Root Authority

Root authority is called domain authority data owner is created by root authority. Root authority creates private key corresponding public key for all data owner of enterprise. Public key publishes the key to all parties in the enterprise and private is kept secret and known only authorize party. Root authority is associated with following database tables namely data owner information table, private key configuration table, symmetric key information table and public key configuration table. The Table1 describes data owner information table. Data owner information table contains individual person information like owner_id, owner_name,owner_ phonenumber,owner _ emailid.owner_password and owner_ file.

TABLE I DATA OWNER INFORMATION TABLE

Owner id	Patient name	password	Phone number	E_mail Address	Owner_Files
1	Sambi	******	9733999088	Sambi123@yahoo.com	manju
2	Priyanka	********	9843322233	priyankamovieon@gmail.com	File2
3	Arjun	********	9879000008	ajrunkid@yahoo.com	File3
4	Akshra	*********	9870777777	Akshrareddy1@gmail.com	File4
5	Cherishma	*******	0877774499	Cherishmasam@gmail.com	File5
6	Hanu	**********	9840124778	Hanumantha.apps@gmail.com	File6

The table 2 describe public key configuration table which contains owner id, public key and key date. This table can be accessed by gateway server.

TABLE II Public key configuration Table

Owner id	Public key	Key Date

The table 3 describe Symmetric key information table which contains Symmetric key and key date this table can be accessed by gateway server.

TABLE III Symmetric key Information Table

Symmetric key	Key Date

In table 4 describes the private key configuration table which contains owner id, private key and key date.

TABLE IV Private key configuration Table

Owner id	Private key	Key Date

4.2. Data Owner

Data owner is created by root authority. Data owner creates Data users, File creation, File deletion, File Modification and Message digest for file is created

4.2.1. Creating Data User

Data owner is created by root authority. Secret information file is created for data users by data owner. The secret information file is created . In this work we set depth of key structure is 3. The secret information file is encrypted using homomorphic encryption and sent to user[8]. Let the secret information file be s and send E(s) to server. Cipher text E(s) at server side can assess a operation f on s and get encrypted output is E(f(s)).The user gives needed functionality and decrypt the results, cloud service provider takes nothing but information that it should be calculated. The homomorphic encryption scheme performs following tasks they are key generation, encryption and decryption. In Key Generation (k) the algorithm takes input as s and gives output as public key and private key[17]. In encryption the algorithm takes plaintext s, private key and public key and produces output as cipher text. In decryption algorithm takes input as cipher text, private key and produce output as plaintext.

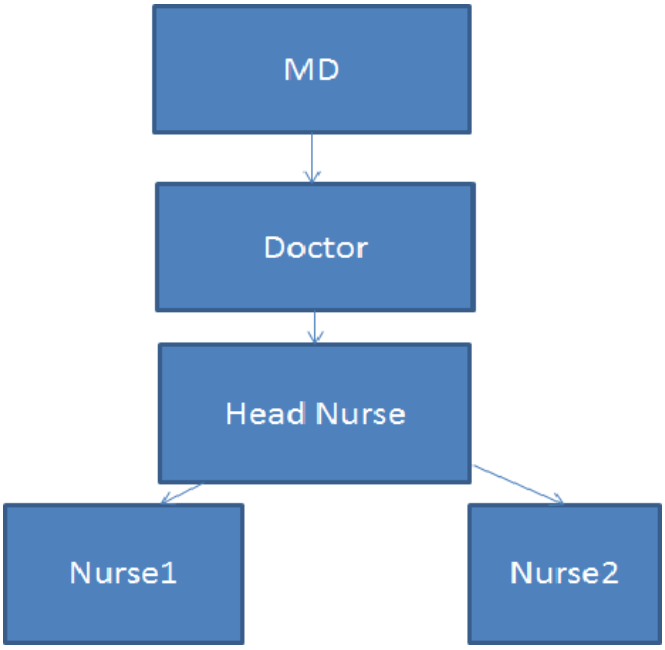

Fig 2:Key structure of Hosiptal Database

4.2.2 Creating new File with message digest

The data owner create the file and select the unique id from file .Message digest with read and write access of particular file is generated by data owner with the help of hash function and stores in database. File is encrypted write access using homomorphic encryption and stores the encrypted file in cloud service provider. Data owner decides the access structure of the created file [9]. The access structure describes whether the user has rights to access the file and have access to download file or not. MD has right to access entire data of Doctor, Head nurse and Nurse. Doctor has right to access the data of Head Nurse and Nurse.Head Nurse have right to access data of Nurse. Doctor has right to access their own data[10]. The table 5 describes the information of the file which contains Owner id, file name, message digest and cloud service provider.

TABLE V File Information Table

Owner id	File name	Message Digest	Cloud Service Provider
1	Manju.txt	23420909780fg4ghtrgft 4787879897989	Cloud 1

4.2.3. File deletion by data owner

Data owner can delete the encrypted file by requesting cloud service provider. Data owner send the request with file id to cloud service provider. Cloud Service Provider deletes the file by verifying data owner and then deletes the file.

4.2.4. File access by user

File can be accessed by user based on access policy. User can access the file depending on access policy .The table 5 describes the file access information table which contains access policy code, file name, employee project and employee designation.

5. Performance of Homomorphic Hierarchical Attribute based encryption is compared with key-aggregate cryptosystem (KAC) and Hierarchical Attribute Based Encryption in Cloudsim

The proposed algorithm is implemented in clouldsim using NetBeans. Cloudsim required some components for implementation they are Cloudlet, Datacenter, Datacenter Broker, Host, VmAllocationPolicy, VmSchedulerPolicy and virtual Machine

Cloudlet –Incoming job from user for operating system.

Datacenter - Represents data owner.

Datacenter Broker- Broker is mediator between user and datacenter.

Host –For Hosting virtual Machine host is used.

VmAllocationPolicy –Allocation of virtual machine to host in Datacenter by policy

VmSchedulerPolicy- policy used for sharing processing power among VMs, running in a host.

The performance of proposed approach is compared with existing approach based on data integrity, confidentiality and misbehaviour

5.1. ***Data Integrity of Homomorphic Hierarchical Attribute based encryption is compared with key-aggregate cryptosystem (KAC) and Hierarchical Attribute Based Encryption (HABE) is measured in terms of security***

Data integrity is refers to quality of data in database. Example of data integrity is if the table has patient_id column value of 123, the database should not allow another patient to have an ID with same value and if in the database having an doctor rating column have values ranging from 1 to 10,the database will not accept a value 11.In Homomorphic Hierarchical Attribute based encryption archives data integrity using message digest[17] .The message digest guarantees that the message has not been altered. The proposed

Homomorphic Hierarchical Attribute based encryption provides better data integrity of about 16-26% compared with key-aggregate cryptosystem 11-13% and Hierarchical Attribute Based Encryption (HABE) 6-11%.We achieve better data integrity in Homomorphic Hierarchical Attribute based encryption because read and write request is generated along with message digest .

For example we are taking files of 2MB, 4MB, 6MB, 8MB,10 MB,12MB, 14 MB transferring file from file from one data canter to another data canter, during data migration of the file from one one data canter to another data canter data integrity of file is measured based on Homomorphic Hierarchical Attribute based encryption , key-aggregate cryptosystem (KAC) and Hierarchical Attribute Based Encryption (HABE).

The table 6 represents Data Integrity of Homomorphic Hierarchical Attribute based encryption(HHABE) is compared with key-aggregate cryptosystem (KAC) and Hierarchical Attribute Based Encryption (HABE) is measured in terms of security.

TABLE VI Data Integrity with respect with KAC

Scheme, HABE

Scheme , HHABE

No. of Data File (MB)	**Data integrity in percentage**		
	KAC scheme	**HABE scheme**	**HHABE scheme**
2	**36**	**42**	**53**
4	**43**	**46**	**56**
6	**54**	**56**	**68**
8	**48**	**52**	**69**
10	**61**	**64**	**73**
12	**75**	**77**	**86**
14	**70**	**72**	**89**

5.2. ***Confidentiality of Homomorphic Hierarchical Attribute based encryption is compared with key-aggregate cryptosystem (KAC) and Hierarchical Attribute Based Encryption (HABE) is measured in terms of security***

Confidentiality keeps information in database secret and protects the information from unauthorized access Sensitive data is collected by Research agencies of hospi

TABLE VII Confidentiality with respect with KAC

Scheme, HABE

Scheme , HHABE Scheme

Block Sizes (KB)	**Confidentiality in Percentage**		
	KAC scheme	**HABE scheme**	**HHABE scheme**
50	40	46	55
100	32	37	40
150	31	35	56
200	42	48	55
250	48	41	50
300	44	29	34
350	52	26	77

We gather confidentially information of clients such as drug use and HIV status. It is required task to protect the confidentiality information of the clients. Homomorphic Hierarchical Attribute based encryption achieves Confidentiality by two times encrypting data.Frist encrypt the data using Homomorphic attribute based encryption. Second access policies are found based Hierarchical Attribute Based Encryption. Data encrypted using Homomorphic Hierarchical Attribute based encryption is highly confidently compared key-aggregate cryptosystem (KAC) and Hierarchical Attribute Based Encryption (HABE).The unauthorized access handled more potential by Homomorphic Hierarchical Attribute based encryption about 13-26% where KAC scheme is about 7-13% than HABE scheme 14-18%.

5.3. *Validity of Homomorphic Hierarchical Attribute based encryption is compared with key-aggregate cryptosystem (KAC) and Hierarchical Attribute Based Encryption (HABE) is measured in terms of misbehavior*

The data validity is defined as how long data is valid with respect to data owner and data consumer. The validity of data is also refer as accuracy of data. In Homomorphic Hierarchical Attribute based encryption we validate the data with respect to message digest algorithm. Data owner first creates message digest for the file and then encrypt the file and send to cloud provider.

TABLE VIII Confidentiality with respect with KAC

Scheme, HABE

Scheme, HHABE Scheme

No. of Data File (MB)	**KAC scheme**	**HABE scheme**	**HHABE Scheme**
5	**52**	**49**	**65**
10	**37**	**36**	**49**
15	**33**	**31**	**45**
20	**29**	**29**	**40**
25	**41**	**39**	**57**
30	**43**	**41**	**59**
35	**49**	**44**	**64**

Third party auditor takes file from cloud provider generate message digest. Data user accesses the file from cloud provider and encrypts the file and verifies the message digest obtained from data owner and data provider is same. If both message digest is same the data is valid else not valid. Better validity of data provided by Homomorphic Hierarchical Attribute based encryption is about 10-15% compared with key-aggregate cryptosystem (KAC) provides 20-25% and Hierarchical Attribute Based Encryption (HABE) provides 30-35% is measured in terms of misbehaviour.

6. CONCULSION

Homomorphic Hierarchical Attribute Based Encryption with integrity proof is proposed in this paper to avoid security risk in cloud. Homomorphic Hierarchical Attribute Based Encryption is compared with key-aggregate cryptosystem (KAC) and Hierarchical Attribute Based Encryption (HABE) based on data integrity, confidentiality and Validity. The Homomorphic Hierarchical Attribute based encryption provides better data integrity of about 16-26% compared with key-aggregate cryptosystem 11-13% and Hierarchical Attribute Based Encryption (HABE) 6-11%.. Homomorphic Hierarchical Attribute based encryption is highly confidently compared key-aggregate cryptosystem (KAC) and Hierarchical Attribute Based Encryption (HABE).The unauthorized access handled more potential by Homomorphic Hierarchical Attribute based encryption about 13-26% where KAC scheme is about 7-13% than HABE scheme 14-18%.Better validity of data provided by Homomorphic Hierarchical Attribute based encryption is about 10-15% compared with key-aggregate cryptosystem (KAC) provides 20-25% and Hierarchical Attribute Based Encryption (HABE) provides 30-35% is measured in terms of misbehaviour. Homomorphic Hierarchical Attribute Based Encryption provides better data integrity, confidentiality and Validity compared with key-aggregate cryptosystem (KAC) and Hierarchical Attribute Based Encryption (HABE) based on data integrity, confidentiality and Validity.Security Level of Homomorphic Hierarchical Attribute Based Encryption is high compared with existing techniques

7. References

[1] R.Manjusha,R.Ramachandran," Highly Secured Cloud Computing using Functional Encryption Scheme",Information journal,japan ,vol.17,no.9(b),septemper ,2014.

[2] R.Manjusha,R.Ramachandran," Sharing Data In Cloud Based On Trust Attribute Based Encryption (Tabe)", ARPN Journal of Engineering and Applied Sciences, ISSN 1819-6608, VOL. 10, NO. 9, MAY 2015

[3] Shucheng Yu,Cong Wang and Kui Ren,"Attribute Ba,sed Data Sharing with Attribute Revocation",ASIACCS'10 April 13–16, 2010, Beijing, China.

[4] Pallavi R and Dr. R Aparna,"Ensuring Integrity Proof in Hierarchical Attribute Encryption Scheme using Cloud Computing",International Journal of Cognitive Science, Engineering, and Technology Volume 1, Issue 1, November-2013 ISSN 2347 – 8047.

[5] Eman M.Mohamed, Hatem SAbdelkader and Sherif EI-Etriby,"Enhanced Data Security Model for Cloud Computing",The 8th International Conference on INFOrmatics and Systems (INFOS2012) - 14-16 May Cloud and Mobile Computing Track.

[6] Guojun Wang, Qin Liu , Jie Wub and Minyi Guo c,"Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers",http://trust.csu.edu.cn/faculty/csgjwang available at www.sciencedirect.com journal homepage: www.elsevier.com/locate/cose 2011 Elsevier Ltd.

[7] Danan Thilakanathan, Shiping Chen, Surya Nepal and Rafael A. Calvo,"Secure Data Sharing in the Cloud", Security Privacy and Trust in Cloud Systems, 45 DOI: 10.1007/978-3-642-38586-5_2, © Springer-Verlag Berlin Heidelberg 2014.

[8] Michael Brenner, Jan Wiebelitz, Gabriele von Voigt and Matthew Smith ," Secret Program Execution in the Cloud Applying Homomorphic Encryption", IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2011), 31 May -3 June 2011,ISBN: 978-1-4577-0872-5 (c) 2011 IEEE.

[9] Dan Boneh and Xavier Boyen. Efficient selective-id secure identity based encryption without random oracles. In Proc. of EUROCRYPT 2004, volume 3027, LNCS, 54–73. Springer.

[10] Cheng-Chi Lee1, Pei-Shan Chung, and Min-Shiang Hwang,"A Survey on Attribute-based Encryption Schemes of Access Control in Cloud Environments", International Journal of Network Security, Vol.15, No.4, PP.231-240, July 2013.

[11] Amna Ahmed Ali1 ,1Kenana Sugar Company, " A Comparative Study of Fully Homomorphic Encryption Schemes for Cloud Computing" International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Special Issue 4, October 2013

[12] Dr.A.Padmapriyam and P.Subhasri, ," Cloud Computing: Security Challenges & Encryption Practices", International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X, Volume 3, Issue 3, March 2013.

[13] Sascha M¨uller, Stefan Katzenbeisser, and Claudia Eckert," Distributed Attribute-Based Encryption", ICISC 2008, LNCS 5461, pp. 20–36, 2009 Springer-Verlag Berlin Heidelberg 2009.

[14] Amazon elastic mapreduce. See http://aws.amazon.com/elasticmapreduce

[15] Wenjuan Li and Lingdi Ping, "Trust Model to Enhance Security and Interoperability of Cloud Environment", Proc. of CloudCom 2009, Springer- Verlag Berlin Heidelberg 2009, LNCS 5931, pp. 69–79, 2009.

[16] M. Abdalla, E. Kiltz, G. Neven, Generalized key delegation for hierarchical identity-based encryption, in: ESORICS'07, Springer Berlin Heidelberg, 2007, pp. 139-154

[17] S.Shanawaz Basha And A.D.Sivarama Kumar," Implementation Of Security Services By Using Sla In Multicloud Computing", Journal Of Information, Knowledge And Research In Computer Engineering.