Volume 5, Issue 6, June 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Secure Self-Governing, Synchronic and Role-based Access to Encrypted Cloud Database

Sharvari A. Pawar*, S. B. Rathod
Department of Computer Engineering,
Sinhgad Academy of Engineering,
University of Pune, India

Abstract— Cloud computing- an expeditiously growing area. It's observed continuous growth in trend of utilization of cloud for usage of massive database storage. This has resulted in increase of security related issues to control and/or to avoid unauthorized accessibility of data/database that is stored on cloud. Various encryption techniques and algorithms are used to facilitate confidentiality. Drawback is only cryptographic techniques are not satisfactory. This paper sketches an architecture where techniques of cryptography in combination with Access control policies are utilized strengthen security. In this paper, you will observe use of Role-Based Access control model. Here, user to role and role to privileges are port-rated. Study about combination of encryption and RBAC model is shown in this paper. RBAC model here is plotted on the encrypted data/database in cloud.in this type of architecture the data of an organization or enterprise is securely stored on public cloud and crucial data of organization or enterprise along with roles and their privileges are stored in private cloud. The organization's data is accessed as per the designation (role) of the user so as to assist in conserving data security in cloud domain.

Keywords— Cloud computing, Security, Cryptographic algorithms, Role-based access model.

I. INTRODUCTION

Today internet has turned the world into global village where news; data, etc. are available in real time to the consumers. The piling up of this huge amount of data is a matter of concern. Thanks to our saviour, the 'CLOUD' where huge amount of data can be stored. It provides us with instantaneous availability of storage bandwidth, assured disaster data recovery, automatic security updates, flexibility to work from anywhere and using any device (as data can be made available by the cloud service providers using internet). Cloud is classified based on the ownership of data and its availability into Public Cloud, Private Cloud, Hybrid Cloud and Community Cloud.
Further cloud also provides different services to its users such as,

A. Infrastructure-as-a-Service

The user can get computing infrastructure, if he demands, from the service providers, which is chargeable [3].

B. Software-as-a-Service

The user can also demand software from the service providers.
Ex: Common business application like SCM, CRM, etc. is delivered through this model by providers like salesforce.com, etc [5].

C. Platform-as-a-Service

It provides a computational platform for the user where he can display the code and the faster task from storage to computing is done by the cloud [4].

Few issues of cloud computing are Data security, Charging model, SLA, migration, Cloud interoperability Issues and Costing model [6]. This paper majorly focuses on the security issues [7] in cloud. Kaspersky Security solutions of Russia recently surveyed the cyber world and inferred that countries like US, China, India etc. are more prone to cyber threats. This cyber security remains a major concern in computing world.

II. RELATED WORK

For strengthening security we are trying to combine the encryption techniques along with access control policies. Sugumaran et at [8] has used block symmetric cryptography for encrypting data for cloud storage. The data to be encrypted is divided in same size blocks and then encrypted and stored in cloud. Key used for encryption and decryption is the same. Advantage of this architecture is its speed of sorting the data, but on the other hand as same keys are used the unauthorized can easily get the key and also the hold of data as well.

Maha TEBBA et al [9] proposed and architecture which uses Homomorphic encryption for the data to be stored. Homomorphic means actions can be performed on the encrypted data without decrypting it. Neha Jain et al [10] has proposed a technique to secure cloud data using DES algorithm. The data encryption standard algorithm takes plain text and key as input and provides us with cipher text. Architecture is mainly based on DES cipher block chaining. No doubt this technique will secure cloud data, but we need to step up to enhances and strengthen it. Monikandan et al [11] has given an architecture which uses classical encryption techniques i.e. both substitution and transposition. Here same key is used for encryption and decryption. Plaintext is converted into ASCII code value, and key ranges from 1 to 256. This classic encryption encrypts the data and stores onto the cloud, it can't be accessed by cloud provides as well as attackers.

The Access control problem can be altered to as key management problem. The key management in [12]-[14] is built on hierarchical key management, so as to establish RBAC policies on architecture. In HKM, keys for each file of data need to be kept safe by the user. But if there are large numbers of uses involved the overhead of managing so many keys will be hectic. Drawback is if role of certain users change, all keys related to that user need to be changed and this will be more hectic job.

In Hierarchical ID-based encryption (HIBE) [15]-[16] the ancestor node can retrieve the keys for encryption from the identities of their child nodes. It is a hierarchical structure with each node having its identity, and the identity of child node must be the subset of the parent node. But if large numbers of users are involved, the length of identities will increase with the depth of the hierarchical tree. Major drawback is if a certain node is assigned to any other parent node, all the identities need to be altered.

There are many Access Control Models such as RBAC, MAC and ABAC etc. In ABAC, each authorized user is granted access based on its attributes. The system defines certain combination of attributes to access a certain file. If a user wants to access a file he need to prove that he has all the attributes needed for access. Using this model, ABE scheme was proposed [17].In this, data encryption is done using set of attributes, and the users having private key aligned with these attributes can decrypt the particular data.

III. PROPOSED SYSTEM

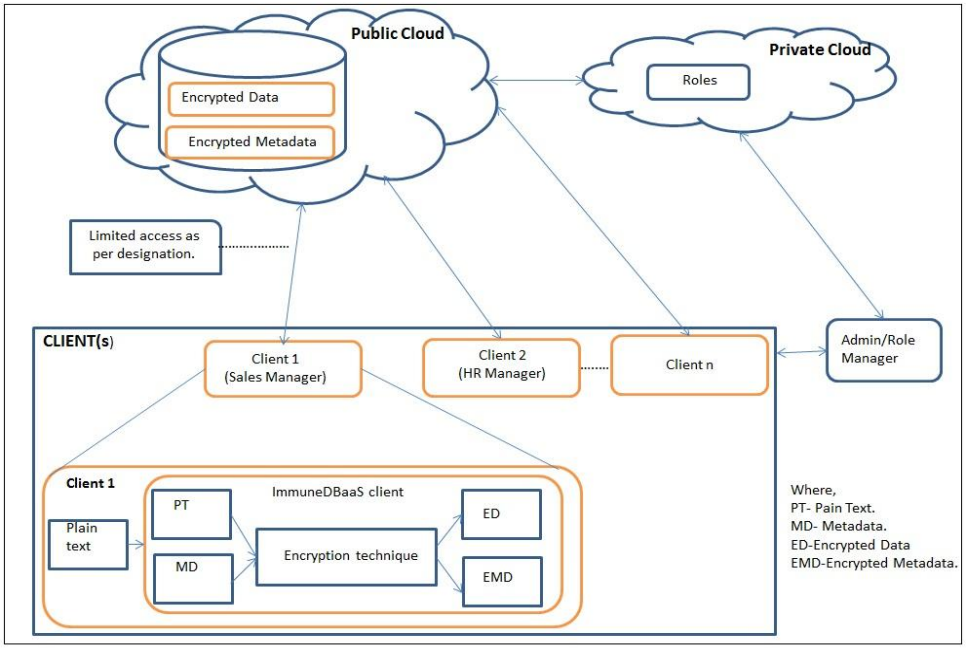

Fig 1. Proposed System

A. Architecture:

ImmuneDBaaS is framed to provide access to multiple authenticated clients with cloud database in synchronic and self-governed manner [1]. These clients have restricted access to the cloud database as its stated designation in organization. System overview diagram explains the architecture in detail. Let's examine an organization that wishes to place its crucial database on cloud storage. In each of the machine the occupant then establishes ImmuneDBaaS. Later the users are provided access to the cloud database by these clients. Each Client has accessibility to upload the contents as well as access the available contents from cloud database. SQL operations are carried out for accessing the database. Accessibility to the database is provided as their stated designation [2].

Some of the major elements at ImmuneDBaaS client side are as follows

1) *Plaintext data:* Database in plaintext form which the client wants to upload in cloud.
2) *Metadata:* It contains information about the file which the client wants to upload.
3) *Encrypted data:* Using any of the encryption algorithms such as AES, Triple DES, Blow Fish etc. the plaintext database is converted into encrypted database, which is then stored onto cloud by the client.
4) *Encrypted metadata:* Using encryption algorithm the metadata is converted to encrypted metadata. Along with encrypted database, metadata is also stored onto cloud.

B. Components:

1) Public Cloud: Public cloud is a publically available to all via browser. Third party is entirely responsible for its creation and preservation. Plaintext database can't be uploaded directly onto cloud. If uploaded directly, any unauthorized person can access it and misuse it. So, only general information and encrypted database of organization is uploaded in public cloud.

2) Private Cloud: Private cloud is built by an organization privately. Confidential and vital data/database of organization is stored in private cloud. To ensure and strengthen security of the data no client (user) can directly access private cloud. Reducing the surface of attack on private cloud. Only Admin/Role manager has been given an interface to the private cloud.

3) Client(s): Clients can either be a user or Owner of the data. Clients as user want to access a certain data from public cloud. Owner as user wants to store his data on the cloud in encrypted form. Owner is the one who decides which all roles should be given access for that particular file. Also defines the permission's (Read, Write, and Delete) for each role. Each client is authenticated by admin, using one of the authentication mechanisms. After successful authentication, the client specifies weather he wants to upload a file or access a file.

4) Admin/Role: Manager- Admin is the authentication manager who takes care of authentication policies. Provides role to each client after a new registration. Roles in role hierarchy are created and managed by admin in private cloud. Role parameters if changed are updated by admin/role-manager. Admin has interface with private cloud.

C. Operations

Assume the system uses a secure encryption scheme to encrypt messages using the private key.

1) Encryption: Initially, Client wants to upload the database on cloud. This data is passed to ImmuneDBaaS Client associated with the respective client. It then extracts the metadata from the database, and converts the database and metadata to encrypted database (ED) and encrypted metadata (EMD) respectively using certain encryption techniques. Database is encrypted using private key and metadata is encrypted using master key. Master key is provided by the admin to the clients while approving there registration. While uploading the owner decides which groups should be given access and as per specifications the admin manages the roles in private cloud.

2) Storage in Cloud: The ImmuneDBaaS Client then uploads the ED and EMD in cloud.

3) Access: If a certain Client wants to access a certain data/database, he requests for the EMD from the Public cloud, the public cloud then passes the Client ID and the file name he wants to access. In private cloud, verification is performed, whether this client has access or not.

i. Has access: The public cloud sends the encrypted metadata of that file so that client will be able to access it. Using this encrypted metadata, the client frames the SQL query in encrypted form. This query is then fired to Public cloud. The SQL query (Select statement) is then performed on the CloudDB.

ii. No access: The public cloud will return a message that no access to this user.

4) Decryption: The database requested is send in encrypted form to the particular client, then at the client side this file is decrypted using private key.

IV. CONCLUSION

In this paper we have outlined an architecture, which combines the encryption techniques and Access control model to toughen the database security in cloud environment. Role based access control model is used in this architecture. In this architecture, the clients can access the database based on their role in that organization.

ACKNOWLEDGMENT

First and foremost, I would like to thank our P.G. coordinator**, Prof. S. N.Shelke,** for his guidance and support. We will forever remain grateful for the constant support and guidance by guide. Through our many discussions, he helped us to form and solidify ideas. The invaluable discussions we had with him, the penetrating questions he has put to us and the constant motivation, has all lead to the development of this paper. I wish to express my sincere thanks to Head of the department **Prof. B. B. Gite.** I would also like to thank to my friends and supporters for listening to our ideas, asking questions and providing feedback and suggestions for improving ideas.

REFERENCES

[1] Luca Ferretti, Michele Colajanni, and Mirco Marchetti "Distributed ,Concurrent, and Independent Access to Encrypted Cloud Databases",IEEE transactions on parallel and distributed systems, VOL. 25, No. 2, February 2014.

[2] Lan Zhou, Vijay Varadharajan, and Michael Hitchens , "Achieving Secure Role-Based Access Control on Encrypted Data in Cloud Storage", IEEE transactions on information forensics and security, VOL. 8, No.12, December 2013.

[3] Amazon elastic compute cloud web services.http://aws.amazon.com/ec2.

[4] Netsuite saas portal. http://www.netsuite.com.

[5] Salesforceforce.com platform. http://developer.force.com.

[6] Kuyoro S. O, Ibikunle F. and Awodele O., "Cloud Computing Security Issues and Challenges,"International Journal of Computer Networks (IJCN), VOL. 3, No. 5, 2011.

[7] W. Jansen and T. Grance, "Guidelines on Security and Privacy in Public Cloud Computing,"Technical Report Special Publication.

[8] Sugumaran, BalaMurugan. B, D. Kamalraj," An Architecture for Data Security in Cloud Computing", IEEE World Congress on Computing and Communication Technologies 2014.

[9] Maha TEBAA, Sad EL HAJJI, Abdellatif EL GHAZI "Homomorphic Encryption Applied to the Cloud Computing Security" Proceedings of the World Congress on Engineering, London, U.K., Vol 1, July 4 - 6, 2012.

[10] Neha Jain and Gurpreet Kaur," Implementing DES Algorithm in Cloud for Data Security" VSRD-IJCSIT, Vol. 2 (4), 2012, 316-321.

[11] L. Arockiam, S. Monikandan," Data Security and Privacy in Cloud Storage using Hybrid Symmetric Encryption Algorithm", International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 8, August 2013.

[12] S. G. Akl and P. D. Taylor, "Cryptographic solution to a problem of access control in a hierarchy," *ACM Trans. Comput. Syst.*, vol. 1, no. 3,pp. 239–248, 1983.

[13]] M. J. Atallah, K. B. Frikken, and M. Blanton, "Dynamic and efficient key management for access hierarchies," in *Proc. ACM Conf. Comput. Commun. Sec.*, Nov. 2005, pp. 190–202.

[14] H. R. Hassen, A. Bouabdallah, H. Bettahar, and Y. Challal, "Key management for content access control in a hierarchy," *Comput. Netw.*, vol. 51, no. 11, pp. 3197–3219, 2007.

[15] C. Gentry and A. Silverberg, "Hierarchical ID-based cryptography," in *ASIACRYPT* (Lecture Notes in Computer Science), vol. 2501. New York, NY, USA: Springer-Verlag, 2002, pp. 548–566.

[16] D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical identity based encryption with constant size ciphertext," in *EUROCRYPT* (Lecture Notes in Computer Science), vol. 3494. New York, NY, USA: Springer- Verlag, May 2005, pp. 440–456.

[17] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. ACM Conf. Comput. Commun. Sec.*, Oct./Nov. 2006, pp. 89–98.

