AN EFFICIENT ROLE-BASED ACCESS CONTROL ON ENCRYPTED DATA IN CLOUD

R.Vijayakumar[1], D. Vijayakumar[2], Dr. K.G.Srinivasagan[3]

[1]PG Scholar, [2]Assistant Professor, [3]Professor and Head, Department of CSE – PG, National Engineering College, Kovilpatti, Tamilnadu, (India)

ABSTRACT

Cloud Computing is a virtualized compute power and storage delivered via platform-agnostic infrastructures of abstracted hardware and software accessed over the Internet. Cloud Computing involves delivering hosted services over the internet. These services are broadly categorized into three types, Infrastructure-as-a –Service (IaaS), Platform-as-a-Service(PaaS) and Software-as-a-Service(SaaS). It is a shared pool of resources, multiple users can share their information via cloud, and security is the main issue for use and shares their data. In this proposed work, an Efficient Role Based Access Control technique uses a public key cryptosystem is considered. First phase involves a user Identity Registration and Identity Token issuance and Role Based Key Generation for all the users assigned to a roles in a system. Second phase involves Data Encryption, along with roles and TokenID for that file, data are encrypting with appropriate role public keys. Third phase involves a Data Decryption, in which role based on their access policies can decrypt the data, and it is done by using their private key. This proposed work deployed in Cloud-Stack private cloud (IaaS). The entire system will enhance the confidentiality and authentication for the cloud users.

Keywords: Cloud Computing, Data Security, Encryption, Access Control

I. INTRODUCTION

The rapid development occurring in cloud computing and services, there has been an enhancing trend to use the cloud for large-scale storage. This has raised the important security issue of how to control and prevent unauthorized access to store in the cloud. Mainly, the healthcare industry needs the efficient storage and quick sharing of electronic medical records with other healthcare and government agencies with efficient security. Lack of Security in the cloud environment is the way to get or shown a sensitive data by unprivileged user or unauthenticated user in that environment Data confidentiality is also a main issue in cloud data.

In this proposed work, an efficient role-based access control on encrypted data is considered in cloud environment. A efficient data security for owners data is achieved by two level authentication for user and key generation along with his role assigned in the system and mainly tokenId is created for every file. An access rights are defined for their roles. We use a public key cryptosystem for generate a public key and private key for roles are in a system. This public key cryptosystem avoid the main problem of key escrow problem in cloud environment. Data owners should be able to assign other cloud user which means already registered users with different access privileges to their data. To enforce these access rights, this scheme defines a public-private key pair for each role. Data files are encrypted by public key. Private key is used to decrypt the file. The main idea is user identity is fully hidden from the cloud environment

and authenticated users are involved in this system. Access rights are also considered for every roles assigned in the system. The Major contribution of this paper is listed below.

- The user identity token registration are used generate a unique Id for registered users. This unique Id and role based keys are used create a efficient two level authentication.
- Data encryption is achieved by key of each role. Use a public key cryptosystem generate a public key and encrypt the data file and generated tokenID of the file.
- Data decryption is achieved by user private key. With this private key users role identity is attached. The role identity is permit the what kind access rights they have and give permission for that encrypted data file to get decrypt the file.

The following sections of the paper are organized as follows. Section 2 introduces the related works of this paper. Section 3 presents our role based key generation algorithm. Section 4 evaluates and compares the performance of the proposed work with other existing algorithms. Lastly, Section 5 concludes the paper.

II. RELATED WORK

Various researches works in the field of data privacy and security in cloud computing system were performed. Those research works based on various categories are summarized as follows.

One of the works considers two layers of encryption for cloud data; under this approach the data owner performs a coarse-grained encryption, whereas the cloud performs a fine-grained encryption on top of the owner encrypted data. It minimizes the overhead at the data owners and computation cost. This system assures the confidentiality of the data and preserves the privacy of users from the cloud while delegating most of the access control enforcement to the cloud. This approach, referred to as Attribute Based Access Control(ABAC), it supports the which is crucial for high assurance data security and privacy. ABAC[1][5] supports fine-grained access control which is crucial for high-assurance data security and privacy. The key problem in this regard is how to decompose(ACP) Access Control Policies, so that the owner handle a minimum number of attribute conditions while hiding content from the cloud. The policy decomposition problem is NP-complete and provided approximation algorithms. This proposed approach to privacy preserving fine grained access control to data in public clouds.

Few work describes new public-key cryptosystem that produce constant-size cipher texts [2] such that efficient delegation rights for any set of cipher texts as possible. The novelty is that one can aggregate any set of secret keys and make them as compact as a single key, but encompassing the power of all the keys being aggregated. In this approach, they consider how to “compress” secret keys in public-key cryptosystem which support delegation of secret keys for different ciphertext classes in cloud storage and more flexible than hierarchical key assignment which only save spaces if all key-holders share a similar set of privileges.

Some related works utilizing ciphertext policy attribute based encryption(CP-ABE)[3] combined with identity-based encryption(IBE) techniques. This policy succeeds in preserving the privacy of cloud users and supports efficient and dynamic operations including, but not limited to, file creation, user revocation and modification of user attributes. CP-ABE[1][3] uses the access structure to encrypt the ciphertext and the secret key is generated based on an attribute set. This approach ensures fine-grained data access control, backward secrecy and security against collusion of users with the cloud and supports user addition, and attributes modification which are not provided by current works. Moreover this scheme does not disclose any attribute of users to the cloud so that keeps the privacy of the users away from the cloud.

Another category based on a novel patient-centric framework and a suite of mechanisms for data access control to Personal Health Record(PHR)s[4] stored in semitrustedservers. To achieve fine grained and scalable data previous works in secure data outsourcing. And this system divide the users in a PHR system into multiple security domains that greatly reduces the key management complexity for owners and uses. This approach gives a high degree of patient privacy is guaranteed simultaneously by exploiting multiauthority Attribute Based Encryption(ABE)[1][3]. This framework addresses the unique challenges brought by multiple PHR owners and users, in that we greatly reduce the privacy guarantees. This approach utilize ABE to encrypt the PHR data, so that patients can allow access not only by personal users, but also various users from public domains with different professional roles, qualifications and affiliations.

Like previous work cloud health information system technology architecture(CHISTAR)[5] that achieves semantic interoperability through use of a generic design methodology which uses reference model that defines a general purpose set of data structures and an archetype model that defines the clinical data attributes. This CHISTAR adopts a two level modeling approach for achieving semantic interoperability. The data integration of CHISTAR allows aggregating healthcare data from disparate data sources. CHISTAR supports security features and address the key requirements of HIAA and HITECH. This approach has better interoperability, scalability, maintainability, portability, accessibility, and reduced costs as compared to traditional client-server HER systems. And also develop a data thinning and progressive sampling approach within the CHISTAR infrastructure that will further improve the querying efficiency and accuracy.

The main contribution of this work is a security architecture that provides a flexible security as a service model that a cloud provider can offer to its tenants and customers of its tenants. It describes the design of the security architecture and discuss how different types of attacks are counteracted by this system. The security policies in TSAD[7][1][6] are enforced only with the consent of the tenants and hence the cloud service providers are not solely responsible for the false alarms due to security policies in the TSAD. This security as a service model while offering a baseline security to the provider to protect its own cloud infrastructure also provides flexibility to tenants to have additional security functionalities that suite their security requirements.

An access control requirement analysis for cloud computing and identifies important gaps, and also proposes an access control[8] model to meet the identified cloud access control requirements. It provides not only ensure the secure sharing of resources among potential untrusted tenants, but also has the capacity to support different access to the same cloud user and gives users the ability to use multiple services securely. In this model, users will be located on a security domain that relates to their role[6]. Every role within the model will be assigned a set of the most relevant and needed tasks for practicing that role. This model secure access and flow of data by marking the data with security labels, which states the data sensitivity. A risk engine is utilized to deal with dynamic and random behaviours of users. A security tags engine is also used for issuing security tags in semi trusted environments and processes. This approach is going to implement an authentication mechanism that can deal with high time and huge space complexity.

we propose a role based access control on encrypted data with efficient two level security of user. The main objective of this project is to provide higher security to the user information and data in the cloud environment. Users do not know where their data is stored and there is a strong perception that users have lost control over their data after it is uploaded to the cloud. This work mainly secure from phishing attack and key escrow attack. In order to allow users to control the access to their data stored in a cloud, suitable access control policies and mechanisms are required..

III. PROPOSED WORK

In this section the system model and the three phases of the entire work is presented in detail.

3.1 System Model

In this proposed work, the role based encryption and decryption are performed in cloud environment. The system model is shown in Fig 1. It is composed of various components like Data owner, User, Administrator and Role Manager. The users are registered their details on user assignment component in the architecture. All the registered user is only can share their information in this system. Role manager manage the all details of what kind role assigned to user in a system and mainly generating the key for appropriate roles in the cloud. Mainly the administrator have the sensitive information like role and their keys and maintain the complete database of the all information involved in this work. In user assignment Fig 2,User of this model or framework is assigned to the Roles. Roles are prepared for his or her responsibility in a system. Mainly all users in system to assigned a appropriate role for further activities. Sometime the role may assign user with his responsibilities and qualifications. First al the users are registered with their details for efficient authentication purpose.

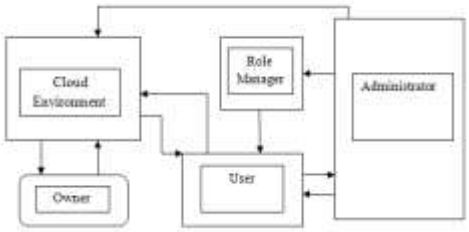

Fig: System Model of Role Based Access Control

The permission assignment is mainly for roles in system. First, the user is assigned to appropriate roles, and then permission is assigned for appropriate roles. Permission are different level in a single role, because a single user have a multiple roles in a system. So permission assignment is important for roles to access the information for appropriate current role. The roles in a system, some roles are inherited a permission of other roles. The admin role has all permissions, like other main roles have more than a single permission, as well as they use lower level user roles.

3.2 Role Based Access Control

Our proposed work involves three phases such as User registration; role based key generation, Data Encryption and Data Decryption. They are discussed briefly below.

3.2.1User Registration and Role Based key generation

In this phase, the user are register their details for involve this system. The user are register their detail for create a user identity token or unique id for every user. This unique id is generating with some attribute of user details. The generated Id is in the form of (1)

$$UniqueID = fl(name) + hashvalue(rn) + fl(DOB) \qquad (1)$$

Where fl is the function gets the first letter or character of the string or word. The name is the user name, rn is the random number generated and DOB is the Date of Birth of the user.

The uniqueID generation for user, use a registering user's first letter of the name , next four digit is generated by unique id generator and last two digit is get from registering user's date in DOB. This user identity token is very unique for all users in the system. After the individual uniqueID generation user can set the password for the user level authentication. Then user can login the system use a user name and password for first level and

security question for user second level authentication. Two level authentications are achieved by this user identity token. Role based key generation is the performed after the registration. Public key crypto system is used for key generation. The roles are assigned in the registration part. In the role attribute is used to generate key. In the public key crypto system, the user public key(e, n, r) is used to encrypt the message and private key(d, n, r') is used to decrypt the message. Role attribute is assigned to both encryption and decryption keys.

3.2.2 Data Encryption

Data encryption phase, the owner wants to encrypt the data with public key. In this phase the owner encrypt the file. With his role based key. We use a some category of roles like, Doctor, specialist, pharmacist and patient, etc. This role are assigned to the user. The patient wants to share the data with other role, here the patient are owner of this data.

In this encryption phase, role based key are used for encrypt the message. Token Id is generated for the data file. This token id is used for the update encrypted file. The access policies are used to give the access rights to the roles. Roles are assigning to the permission with the access rights. In the public key crypto system have the public key(e, n, r). Here 'r' is role attribute in this system. Role hierarchy is created for existing role is a system, which is used to inherit the properties or permission on lower level permission to higher level roles in a system. The Encryption key is of the form given in (2).

$$E_{key} = P_{key} + Role_{ID} + AP \qquad (2)$$

Where the encryption key is E_{key}, P_{key} is the key generated using public key encryption, $Role_{ID}$ is the role identifier and AP is the access policies assigned to the user.

Represent the message as an integer between 0 and (n-1). Large messages can broke up into a number of blocks. Each block would then be represented by an integer in the same range. Encrypt the message by raising it to the eth power modulo n. The result is a cipher text. The resultant cipher text is stored as the encrypted message in the cloud.

3.2.3 Data Decryption

Data decryption phase, user wants to decrypt the message with their private key. Public key crypto system the users need not share the private key, which avoids the key escrow problem. User private key and token id of the file are used to the decrypt the encrypted file. In this phase mainly the user private key (d, n, r') pair with this key pair user role are defined and subtracted from the encryption key or private key. The decryption key is also in the form as shown in (2) and this pattern of key helps in user identification while decrypting the data. In this phase, after the authentication checking the user get the private key. Use the private key and token id of the file user can get the file. Mainly the access rights described for this details. Access rights are like Read, Write, Read Execute, Create file, etc.. The user can only get assigned access control on that data in the cloud. The decryption phase some role have other roles permission from role hierarchy function. Owner file and keys and details are stored in the database securely in cloud environment.

3.2.4 Role Based Key Generation Algorithm

In the Role Based Access Control Model, All the users are mapped to the appropriate Roles. Mainly, In the role based key generation algorithm for generate the key for all users are assigned to roles in a system (Ex : EHR).

- Users are assigned to the appropriate roles in a system. After the Role assignment all the Roles have a particular permission.
- For that Roles, Key is need for the access control permission for user data.

- Use modified RSA is used to create key for Roles.
- We obtain the key for appropriate Role in a system, We going to use any RoleID or Attribute for that Role as Hash value.
- To the generated public key, we going to add some value for that Role. Default value in algorithm is (e, n) here, 'e' is co-prime of the multiplication of two prime number 'n' n= (p-1)(q-1). Publickey(e, n RoleID) for that current user Role.
- For that Role to decrypt or view the original data(information), a private key or a decryption key has to be generated.
- In the encryption key, add the Role with that function. Similarly, Decryption key are also add to default function in RSA. Decryption key(d, n)
- In the Decryption key also have the RoleID details. But the same time users task are performed with this key.
- Before the Decryption key generation, We going to give the access policy for that entering Role. After that create Access policy for the appropriate Roles.
- The decryptor cannot generate the decryption key without AP (access policies).
- AP are generate with Role Attributes. The user satisfies the role attributes for decryption, then allowed permission is granted for that user role.
- AP are used to give access rights of the user Roles for particular permission.

  For our work particular details are seen by many users, so publicly available for users.
- Decryption key is also generated by the user for decryption key before satisfies the access policies.

IV. EXPERIMENT AND EVALUATION

In this section we present the cloud setup and experimental results of our proposed work in detail.

4.1 Experimental Setup

The cloud environment for implementing the proposed system was created using the cloud developing tool called "Cloud-Stack" which provides Infrastructure as a Service (IaaS) for the cloud providers. The Machine 1 acts as the management server that manages cloud resources. By interacting with the management server through its UI or API, we can configure and manage our cloud infrastructure. It controls allocation of virtual machines to hosts and assign storage and IP addresses to virtual machine instances. The Machine 2 acts as the Hypervisor that creates and runs virtual machines and that was managed by machine 1.

4.2 Experimental Results

The experiment was checked with various users and all registration; role based key generation, encryption and decryption. In the first page, the user select the registration button, store the full details of user, mainly the category and security questions which are used for two level of authentication of our proposed work. The roles are assigned from the registration and registered user got the identity token unique id. The every user got the different unique id from random id generator. The user set the password for the first level authentication. The keys generated for the registered user with their file is shown in the Fig 2. The system will ask randomly any security questions to the user registered. This is second level of authentication. If both the level of authentication is successfully completed, then the user is authenticated to use the system. Now user, role based key is generation is generated by the public key crypto system, here which role is entered first his appropriate key is generated.

Fig1: Public and Private Keys Generated

First, the user register their details and generate the user identity token unique id is generated and stored in the cloud database. Unique id is used for further login purpose. The second level authentication is based on security questions. After the registration and login is successful the user is valid user. Role based key was generating for further encryption and decryption. Administrator have the all sensitive information of user. This is shown, only for explain purpose because this information is maintained by the cloud environment.

Fig.2 Decrypted File with its access policies

The access rights are shown in the Fig 4. It shows the doctor role have the access rights Read, Write, Execute.

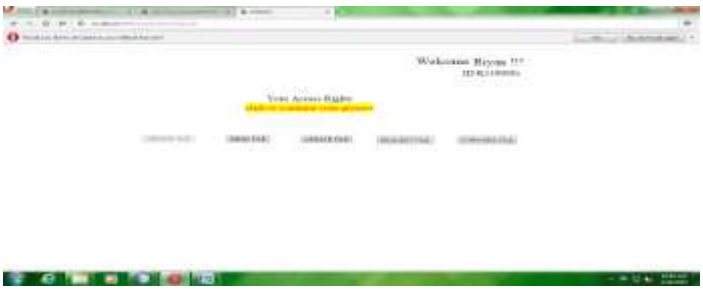

Fig:3 Access Policies for a Doctor

In this work we use a mysql database as the back end to store all details of the user involved in the system.

V. CONCLUSION

An efficient role based access control for enhancing the data privacy and security was proposed. The entire system is deployed in a private cloud configured using Cloud Stack. The system tested with many users of different roles and categories. Each role is assigned with access policies based on their role hierarchy. The unique ID is generated for each user and token ID is generated for all files to be stored in the system. The key generation was done using public key cryptography with the role embedded in it. Finally the decryption key is generated for accessing the file by the user.

REFERENCES

[1] Xin Dong, Jiadi Yu, Yuan Luo, Yingying Chen, GuangtaoXue, Minglu Li (2014), "Achieving an Effective, Scalable and Privacy-Preserving Data sharing service in Cloud Computing", Computer and security 42, pp 151-164.

[2] Cheng-Kang Chu, Sherman S.M. Chow, Wen-GueyTzeng, Jilanying Zhou, and Robert H. Deng (2014), "Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud storage", IEEE transaction on parallel and distributed systems, vol. 25, no. 2, pp 468-477.

[3] Xin Dong, Jiadi Yu, Yuan Luo, Yingying Chen, GuangtaoXue, Minglu Li (2014), "Achieving an Effective, Scalable and Privacy-Preserving Data sharing service in Cloud Computing", Computer and security 42, pp 151-164.

[4] Ming Li, Member, Shucheng Yu, Yao Zheng, Kuiren, and Wenjing Lou (2013), "Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption", IEEE transaction on parallel and distributed systems, vol. 24, no. 1, pp 131-143.

[5] ArshdeepBahga and Vijay K. Madisetti (2013), "A Cloud-Based Approach for Interoperable Electronic Health Records(EHRs)", IEEE journal of biomedical and health informatics, vol 17, no. 5, pp 894-906.

[6] Lan Zhou, Vijay varadharajan and Micheal Hitchens (2013), "Achieving Secure Role-Based Access Control on Encrypted Data in Cloud Storage", IEEE transactions on information forensics and security, vol. 8, no. 12, pp 1947-1959.

[7] Vijay varadharajan, UdayaTupakula (2014), "Security as a Service Model for Cloud Environment", IEEE transactions on network and service management, vol. 11, no. 1, pp 60-77.

[8] Younis A. Younis, KashifKifayat, MadjidMerabti (2014), 'An Access Control Model for Cloud Computing',Journal of Information Security and Applications 19, pp 45-60.