Attribute Based Access Control

Prof.N.B. Kadu[1], Gholap Nilesh[2], Saraf Shashir[3], Garodi Pravin[4], Bora Anand[5]

[1] Asst. Professor, Computer, Pravara Rural Engineering College, Maharashtra, India
[2] UG Students, Computer, Pravara Rural Engineering College, Maharashtra, India
[3] UG Students, Computer, Pravara Rural Engineering College, Maharashtra, India
[4] UG Students, Computer, Pravara Rural Engineering College, Maharashtra, India
[5] UG Students, Computer, Pravara Rural Engineering College, Maharashtra, India

ABSTRACT

Now-a-days as cloud is most widely used in mostly all fields so there is need of keeping data more secure & confidential which is outsourced on the cloud. As the cloud system is decentralized and distributed, any unauthorized user can access this data so to avoid this we need secure access control. In this system the access control is attribute based and ciphertext is used which is better scheme. This system provides hierarchical control structure to reduce burden of central authority. Computation cost of encryption and decryption is less and the size of ciphertext generated is constant. This proposed scheme is efficient, fine-grained and scalable and also increases performance of the system.

Keyword : - *Cipher-text, Attribute based access control, Decentralized.*

1. Introduction

As security is essential in all the fields. Nowadays Cloud Computing has spread so widely that many of the organizations are getting into it rapidly. There are many benefits with cloud computing like reducing capital cost, improving flexibility, disaster recovery, etc. but there exists some unavoidable security problems. These problems need to be concentrated as it may create severe problems which may prevent further development.

In Cloud Computing, users store their data files in cloud server, therefore it is very important to prevent unauthorized access to these resources. In traditional access control methods, it is generally assumed that data owners and the storage server are in the same secure domain and the server is fully trusted. However, in case of the cloud computing environment, cloud service providers may be attacked by malicious attackers due to which the vital private information of users for commercial interests may be leaked as the data owners commonly store decrypted data in cloud servers.

Nowadays, Cloud Computing has attracted widespread attention and support in many fields. Cloud computing can provide several computing capabilities, reduce costs and capital expenditures and charge according to usage. Although there are many benefits, there exist many unavoidable security problems as well. To deal with these problems ABE(Attribute based Access Control) can be the suitable solution , as it is a new cryptographic primitive which provides a promising tool for addressing the problem of secure data sharing and decentralized access control[2] .

Attribute based decentralized access control is on access control paradigm which uses policies containing attributes to grant access to the users through those attributes. ABAC uses attributes as the building blocks to define decentralized access control rules and access requests. In an attribute-based access control system, any type of attribute such as resource attributes and user attributes are used to determine access. These attributes are compared to defined fixed value or even to other attribute, which turns it into a relation-based access control. Attributes come in the key-value pairs such as a "Role=Supervisor," which can be used to limit access to a certain feature of a system. In this case only users with designation of supervisor or higher can be given access to that feature or system. For example, "Permit managers to access financial data provided from finance department". This would allow users with the attributes of Role=Manager and Department=Finance to the access data with the attributes of Category=Financial. This leaves another types of users from even getting to the login page and preventing certain types of attacks like brute force, collusion attack and library attacks.

2. RELATED WORK

Different access control for accessing the data have been proposed from last decades. These schemes like Discretionary, Mandatory, Role-Based Access Control. But these schemes had some limitation and drawbacks so it can’t be used in recent cloud storage systems. Also these traditional access policy has inadequate flexibility and also expansion of these on large scale is more difficult. So there is need to strengthen it’sadaptability.Their adaptability to dynamically change roles is simply not enough. The role of user changes dynamically in many applications. So to achieve dynamic change of role we need a new access control scheme.High security requirements need a new access control model as the traditional access control scheme doesn’t support high level security.To achieve easy public key encryption deployment the concept of identity based encryption was proposed.Ausers public key is his/her identity. An encryptor can create a cipher text under the receiver’s identity without asking for the receiver’s public key beforehand. The first fully functional IB scheme was presented by Franklin and Boneh. Similarly to IBE, a number of identity based cryptographic primitives has been proposed.

Sahai and Water proposed a Fuzzy Identity Based Encryption Algorithm in 2005. The conception of attribute was introduced first and an identity was viewed as a set of attributes in 2006, goyal et al. extended this idea and introduced two variants: - 1) Key Policy and 2) Ciphertext Policy. In a Key policy system, decryption keys are associated with access policies, and cipher text is associated with set of attributes. A user can decrypt cipher text if and only if his set of attributes satisfies the access structure. In CP-ABE the situation is exactly reversed, a user’s private key is associated with set of attributes and encrypted cipher text will specify an access policy over attributes. Various improved ABE algorithms have been introduced and schemes have been presented.

The scheme proposed in this paper is conceptually closer to the traditional access control model such as role based access control model (RBAC).The demerits of this scheme relates to the size of cipher text, and computation of encryption and decryption depends directly on the number of attributes. As there are large number of users in cloud computing environment means it is impractical to complete authorization and distributes secret keys using only single attribute authority. A hierarchical attribute based encryption scheme was proposed by Wang in 2011, to provide full delegation and high performance with fine grace access control. A new versatile crypto system referred to as cipher text policy hierarchical ABE with short cipher text was proposed by Deng.

3. Proposed System

The structure of the system is tree-liked which is formed with root authority, top-level domain authorities and low-level domain authorities to realize attribute management and authority. The structure can divide the burden and risk of the authority of the single central attribute authority in a cloud computing environment. Model proposes a hierarchical CP-ABE access control scheme with constant-size ciphertext. In this size of ciphertext is fix and the computation of encryption and decryption at a constant value for improving the efficiency of the system. The data owner first encrypts the data file using a symmetric key DEK and then encrypts DEK by using the proposed scheme with a specific access control policy. The data owner uploads the final ciphertext and stores it in the cloud servers. Whether a user can access and decrypt the data file depends on how to obtain the symmetric key, which is decided by the user's set of access attributes.

The system model consists of five types of modules:

1) Data owners

2) Users

3) A cloud service provider

4) Attribute Authorities:-

a) Root authority

b) Number of domain authorities.

The big problem in general systemis that ciphertext size depends linearly on the number of attributes but in our systemwe have to use constant size ciphertext which is based on the hierarchical systemmodel.

Proposed systemconsists of 5 algorithms

1) System Setup

2) Domain authority grant

3) Owner/User authority Grant

4) Encryption

5) Decryption

1) System Setup:- Setup($,U)

Root authority calls this algorithm. Using this algorithm root authority can create public key PK and Master key MK.

Input:- Security Parameter $ and Set of all real attributes U.

Output:- Public parameter PK and master key MK.

2) Domain Authority Grant:- CreateDA(MK,PK,A)

If a new domain wants to enter in the systemthen firstly it is authorized by root authority by using attributes of that domain. If attributes are match then domain gives grant to enter into the system.

Input:- Public key PK, master key MK, set of attributes of domain A

Output:- Secret key SK of domain authority.

3) Owner/User Authority Grant:- Delegate(SK,A')

It is used to grant access to new owner/userto enter into the systemaccording to attributes.

Input:- Secret key SK which is generated in Create() algorithm and attributes of new owner/user.

Output:- Secret key SK' of new member

4) Encryption:- Encrypt(PK,M,Y)

Firstly encrypt the data file using symmetric data encryption key DEK and get the ciphertext of data files. Then we encrypt DEK using the CP-ABE algorithm with constant-size ciphertext and obtain the ciphertext of DEK. Therefore users can access the data file by decrypting the ciphertext of DEK and ciphertext of data file.

Input :- Public key PK,message M, threshold access structure Y

Output :- Ciphertext of DEK.

4) Decryption:- Decrypt(Y,PK,SK',CT)

Firstly user decrypts the files by calling Decrypt algorithm to obtain DEK. If attributes are match with threshold access structure then user decrypt CT using secret key SK' to obtain DEK and then decrypt the ciphertext of data using DEK to obtain the original data file.

Input:- Access structure Y, public key PK, secret key SK' and ciphertext CT.

Output:- Key DEK.

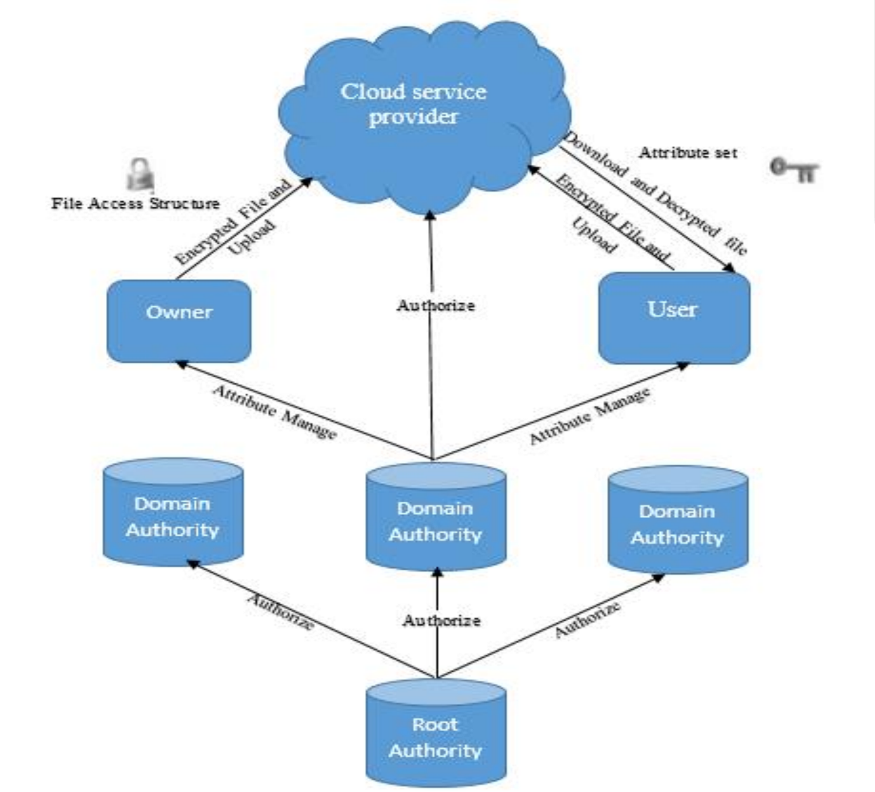

Fig 1: Architecture of System

The cloud service provider manages the cloud servers and provides a data storage service. Data owners encrypt their shared data files and outsource them to the cloud. The format of a stored file in a cloud environment is shown in Fig. 2, where ID is the identity number of a file, DEK is a symmetric key, and CT is the ciphertext of DEK by an ABE algorithm. Since the access structure is implied in ciphertext, only the user with corresponding attributes can decrypt the ciphertext. Unauthorized users cannot access the data file. Hence, we realize access control based on ABE with constant size ciphertext. To access the shared data files, users download an encrypted data file from the cloud and then decrypt the first part of the file CT based on the set of attributes to get the symmetric key. The access policies are expressed in terms of the set of attributes. The user obtains the data file by using the symmetric key to decrypt the ciphertext of the data file. The root authority has the top authority and is responsiblefor generating system parameters and authorizingtop-level domain authorities. Each domain authorityis responsible for managing domain authorities at thenext level or the data owners/users in its domain. Thisinherited structure of attribute authority reduces thecomputation and burden of theauthority of central attribute authority.

This scheme assumes users access the data files in a read-only way. We also assume that the cloud server provider is semi-trusted in the sense which abides by the agreement and faithfully carries out the operating request of a legal user. However it may try to pry into the private files of users or collude with malicious users to harvest file information stored in a cloud for its own benefit. Moreover, we assume communication channels between all parties of a system model are secured.

Performance Analysis

In this, we analyse the performance of proposed system by using graphical representation. As previously mentioned, file’s ID, ciphertext of Decryption Key(DEK) and ciphertext together stored in cloud. But here size of ciphertext is constant, it doesn’t depends upon number of attributes as shown in following fig.

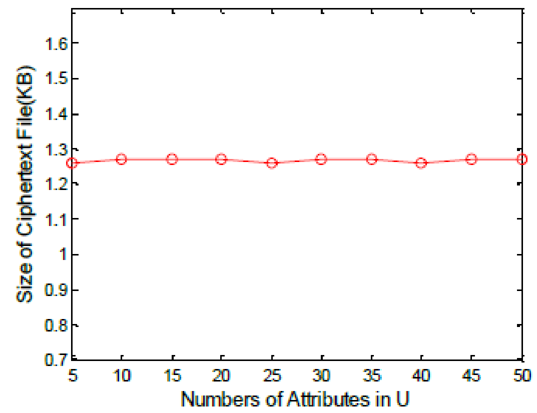

Fig -2: Name of the figure

Introduction related your research work Introduction related your research work Introduction related your research work Introduction related your research work Introduction related your research work Introduction related your research work Introduction related your research work Introduction related your research work Introduction related your research work Introduction related your research work Introduction related your research work Introduction related your research work Introduction related your research work.

FUTURE SCOPE

The above system gives information about secure transfer of data files i.e., text files between different nodes plays an vital role in cloud computing. As we know cloud computing is vast concept. So there are various futurous concepts we can use in this system such as:-

1) Secure transfer of files such as pdf,images,mp3 and video files.

2) In further research, we focus on increasing set of attributes for providing high security.

3) We will try to make system more simple, efficient, scalable and portable.

CONCLUSION

Attribute-based access control provide data confidentiality. This system solves the drawbacks of role-based access control by replacing attributes instead of roles. We use constant size ciphertext instead of depending linearly on no. of attributes which helps to improve efficiency and performance. This shows our scheme has good adaptability and scalability in cloudcomputing.Our scheme can maintain the size of ciphertext and the computationof encryption and decryption at a constant value. Therefore, the scheme can improve the efficiency of the system.

6. REFERENCES

[1] Jianwei Chen and Huadong Ma, "Privacy-Preserving Decentralized Access Control for Cloud Storage Systems" in 2014 IEEE International Conference on cloud computing.

[2] Amazon Elastic Compute Cloud (Amazon EC2). http://aws.amazon.com/ec2/

[3] Amazon Web Service (AWS). http://s3.amazonaws.com/

[4] Z. Wan, J. Liu and R.H.Deng, "HASBE: A Hierarchical Attrib-ute-Based Solution for Flexible and Scalable Access Control in Cloud Computing," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp: 743-754, Apr. 2012.

[5] J. Shao, Z. Cao, "Multi-use unidirectional identity-based proxy re-encryption from hierarchical identity-based encryption",*In-formation Sciences*,vol.206,pp:83–95,2012.

[6] S. Ruj, A. Nayak and I. Stojmenovic, "DACC: Distributed Ac-cessControl in Clouds," *Proc. 10th Int'l Con. Trust, Security and Privacy in Computing and Communications (TrustCom)*, IEEE, pp: 91-98, Nov. 2011.

[7] L. Zhang, Q. Wu, B. Qin, J. Domingo-Ferrer, "Provably secure one-round identity-based authenticated asymmetric group key agreement protocol",*InformationSciences*,vol.181,no.19,pp:4318–4329, 2011.

[8] Z. Liu, Z. Cao, Q. Huang, D. S. Wong, and T. H. Yuen, "Fully securemulti-authority ciphertext-policy attribute-based encryption without random oracles," in *Computer Security–ESORICS 2011*. Springer, 2011, pp. 278–297.

[9] H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure threshold multi authority attribute based encryption without a central authority," *InformationSciences*, vol. 180, no. 13, pp. 2618–2632, 2010.

[10] S. Yu, C. Wang, K. Ren and W. Lou, "Achieving Secure, Scala-ble, and Fine-grained Data Access Control in Cloud Compu-ting," *Proceedings –IEEE INFOCOM*,pp:1-9, 2010.

[11] Google App Engine (GAE). http://code.google.com/appengine/