International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 3, Issue 2, February 2016

Secure Data Destruction In Fuzzy Authorization

[1]Arya Vijayan, [2]Devi Dath
[1] PG Scholar, [2] Assistant Professor In CSE
[1]aryavijayan992@gmail.com, [2] devinirmal05@gmail.com

Abstract- **Cloud computing is a new internet based computing technique that provides many services over internet. Many authorization methods are used to provide security and privacy of data in a cloud. OAuth is one of the most known authorization schemes but cannot be used in the case of heterogeneous clouds. So for the heterogeneous cloud environment Fuzzy Authorization scheme was proposed. This scheme helps in providing the facility for an application in one cloud to access an owner's data that is in another cloud. This is a reading authorization scheme which uses OAuth and modified Ciphertext Policy-Attribute Based Encryption (CP-ABE). If the owner makes any changes in the data, the application's right of accessing the data will be revoked automatically. More security can be given to this scheme by data self destruction method. The implementation is done by using CloudSim and cryptography part is done by Pairing Based Cryptographic(PBC)Library. The simulation result shows that the authorization scheme is more secure and efficient.**

Index Terms— **Ciphertext Policy-Attribute Based Encryption, Key Policy-Attribute Based Encryption, Fuzzy Authorization, Time Specific Encryption, Security.**

I. INTRODUCTION

Cloud computing provides services like Software as a Service(SaaS), Platform as a Service(PaaS), and Infrastructure as a Service(IaaS). Cloud computing has some benefits which includes reduced cost, increased cloud storage[1], flexibility etc. As a result of these benefits huge number of people is now using this facility. This is the main reason why the authorizations and inter working between the Cloud Storage Provider (CSP) and Application Service Provider (ASP) has become important.

Consider the example of an application called PDFMerge[2]. This application is registered to the Google Web Store which is an Application Store (AS). JustCloud is one of the Cloud Storage Provider to which the pdf file of Data Owner (O) is stored. When the data owner needs to merge some of the stored pdf files using PDFMerge application , either direct authorization can be provided to the application by the owner for accessing the files. Another method is that the owner has to first download the pdf files from the cloud storage provider and then upload that to PDFMerge. The first method is more efficient than the downloading and uploading procedures.

To build trust between PDFMerge and data owner that are residing at different cloud parties is an important issue. Another important issue is that if the application needs to provide access right for more than one file, multiple secret keys and access tokens are needed. OAuth[3] is one of the most generally used authorization scheme. This authorization scheme can be used if owner's data and application are in the same cloud party i.e, OAuth scheme cannot be used in the case of heterogeneous clouds. In AAuth authorization scheme, the owner and application service provider are at different clouds. But the problem of multiple access tokens for accessing multiple files is not solved. As a solution to the above mentioned problems Fuzzy Authorization scheme is proposed.

Fuzzy Authorization [5] is a secure file sharing scheme which is highly flexible and scalable. Modified Ciphertext Policy-Attribute Based Encryption [4] and OAuth schemes are used in this scheme. This is a reading authorization scheme. The application has only the right to access the data but has no right to modify the data. Only data owner can make modifications to the data. Another important feature of Fuzzy Authorization is The application's right of accessing the data will be revoked[6] automatically whenever the owner makes any changes to the data.

More security can be provided to this authorization by data self destruction method. This is implemented by using Key Policy-Attribute Based Encryption (KP-ABE)[7]. In this ciphertext will be provided with a time interval and private key is associated with a time instant. The decryption can be done only if the time instant is in that particular time interval and attribute associated with ciphertext satisfy the key's access structure. The sensitive data will be destructed automatically after a user specified expiration time.

II. RELATED WORK

Confidentiality, integrity and access control are some of the issues that are present in cloud storage. There are many access control schemes. This schemes includes Attribute Based Encryption (ABE), Key Policy-Attribute Based Encryption (KP-ABE), Ciphertext Policy-Attribute Based Encryption (CP-ABE).

Attribute Based Encryption was introduced by Sahai and Waters. This scheme provides access control and security. ABE is a type of public key encryption. It allow users to encrypt and to decrypt the data based on their attributes. That is the ciphertext and secret key are based on the attributes. The decryption is only possible if the attributes of secret key matches the attributes of ciphertext. Important feature of attribute based encryption are complex access control and no list of users is needed, just access policy is enough. The main disadvantage of ABE is that to encrypt the data, the owner has to use all the users public key.

The Key Policy-Attribute Based Encryption is a type of ABE. It is designed for one to many communications. In this each private key is associated with an access tree structure. This specifies the type of ciphertext the key can decrypt. Since the ciphertext are represented by a set of attributes and the key is specified using access structure, this scheme is known as KP-ABE. KP-ABE provides more flexibility than ABE. An important problem in this is, the person who is encrypting the data cannot take decision about who can decrypt the data.

The CP-ABE was proposed by John Bethencourt, Amit Sahai, B Waters. This is a type of identity based encryption. It is an important cryptographic primitive for access control. There is one public key and a master private key. The master key can be used to make more restricted private keys. In this scheme the attributes describes the user's credentials and the party who is encrypting the data determines the policy for who can decrypt the data. A user can decrypt the ciphertext only if the attributes of the user passes through the access structure of the ciphertext.

The OAuth scheme proposed by Tassanaviboon provides authorization to an application to access a data when both the data owner and the application resides in same cloud party. This cannot be applied in untrusted cloud. So a new authorization scheme AAuth was proposed. This scheme is based on OAuth standard and CP-ABE. Main advantage of this scheme is that users can share the resources in a secure manner in semi trusted cloud environment. In this for each access grant an ABE token is provided. A modified version of Ciphertext Policy-Attribute Based Encryption and OAuth is used in Fuzzy Authorization scheme.

Time Specific Encryption (TSE) was proposed by Peterson and Quaglia. In TSE there is a time server which broadcast a time instant key. The data owner encrypts the message during a time interval. The receiver can only decrypt the ciphertext if the time instant key is in that time interval.

III. SECURE DATA DESTRUCTION IN FUZZY AUTHORIZATION

A. Concepts

For the secure data destruction in fuzzy authorization based on the KP-ABE with time specified attributes, the following concepts are introduced :

- Expiration Time : It is an instant of time that is defined by the owner. The data can only be done before this instant of time. After this expiration time data will be automatically destructed by itself.
- Authorization Time Period : It is a time interval that is defined by the data owner. The time period starts from the desired release time and ends at an expiration time.
- Lifecycle : It is the time interval from the creation of shared data till the expiration time.

B. System Model

This system consists of 4 entities. They are:

- Data Owner(O)
- Application Service Provider(ASP)
- Cloud storage Provider(CSP)
- Application Store(AS).

Firstly the owner(O) has to register to CSP and then login to it so that owner can store the data by uploading it. Owner can access the stored data and can also provide access right. For processing the data some cloud application services are used. The Application Service Provider needs authorization from the data owner to access owner's data that is stored in cloud storage. PDFMerge is an example for an ASP.

Another entity is the Cloud Storage Provider. The Cloud Storage Provider provides storage as a service. It provides the facility to store data of the data owner. Dropbox is an example for CSP where video and audio files can be stored. Application Service Provider will be registered to an entity called Application Store. Google Chrome Web Store is an example for an Application store. Fig 1 shows the system model.

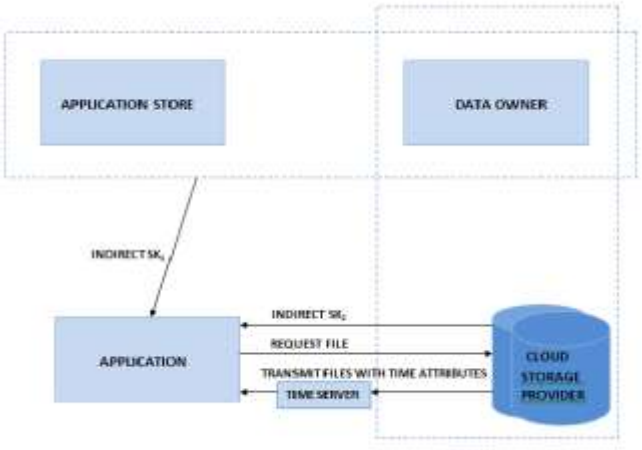

Fig 1. System Model

C. Overview

The authorization scheme has two phases : an offline phase and a running phase. In the offline phase, using a random symmetric key data owner encrypts his data. Then this key is encrypted using modified version of Ciphertext Policy-Attribute Based Encryption and OAuth scheme. The ciphertext of data and ciphertext of key will be encapsulated by the owner to form an archive. The archive is represented by access tree structure and this archive is stored in the Cloud Storage Provider.

In the running phase, when data owner want to share data with an ASP, data owner and CSP will jointly provide an indirect secret share of file attributes and similarly data owner and AS will jointly provide indirect secret share of application attributes. Consider a group element g and secret share s, then the indirect secret share can be represented as g^{sr} . When ASP receives all the indirect secret shares, it will sent request to the cloud storage for the archive and it will be provided to ASP. Decryption of the archive header for the key will be performed. Using this key the ciphertext of data can be then decrypted.

To provide more security to the outsourced data, Key Policy-Attribute Based Encryption with the time specific attribute is introduced. In this each ciphertext is labeled with a time interval and private key is associated with a time instant. Ciphertext can only be decrypted if the time instant is in the specified time interval. The data will be securely self destructed after a user specified expiration time. That is during the encryption time a time interval will be provided for the data. So an authorized application can access the data in that time period only. The data cannot be accessed before the starting time and also after the expiration time. After the expiration time the data will be self destructed. So here a secure data sharing takes place.

III. CONCLUSION

Fuzzy Authorization is a secure file sharing scheme between data owner who stores the data in one cloud and an application which is registered to another cloud. In this scheme confidentiality of data is maintained. It is a reading authorization scheme. Only data owner has the privilege to modify the data. More security is given to the outsourced data by Key Policy-Attribute Based Encryption with Time specified attributes. There is an expiration time, after that instant the data will be automatically self destructed. Thus the data security is established.

REFERENCES

[1] http://www.thetop10bestonlinebackup.com/cloud-storage,

[2] http://www.pdfmerge.com,

[3] A. Tassanaviboon and G. Gong, “Oauth and abe based authorization in semitrusted cloud computing,” in Proc. 2nd Int. Workshop Data Intensive Comput. Clouds

[4] A. J.Bethencourt and B.Waters, “Ciphertext policy attribute based encryption,” in IEEE Symposium on Security and Privacy, pp. 321–334, ACM, 2007.

[5] S. Zhu and G. Gong, “Fuzzy authorization for cloud storage,” in IEEE Transaction on Cloud Computing, pp. 422–436, IEEE, 2014.

[6] C. C. A. O. Michael Backes, “Secure key-updating for lazy revocation,” in Computer Security ESORICS, pp. 41–50, arXiv, 2006

[7] Jinbo Xiong , Patrick s chen,” A secure data self destructing scheme in cloud computing”