Sonam Chugh, Sateesh Kumar Peddoju/ International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 3, May-Jun 2012, pp.2589-2593

# Access Control Based Data Security in Cloud Computing

**Sonam Chugh, Sateesh Kumar Peddoju**
Department of Electronics & Computer Engineering
Indian Institute of Technology Roorkee
Roorkee, Uttarakhand, INDIA

## ABSTRACT

**Cloud Computing is mainly about offering services on pay per user basis. It provides Storage-as-a-Service, where data owner can store their data in the cloud. Data is biggest asset to an organization and how confidentiality, authentication and access control can be outsourced. There is a threat to data owner that if CSP (Cloud Service Provider) is malicious or has some vulnerability. This paper addresses the issue by proposing a framework that secures the documents using hybrid cloud infrastructure with which the data security threat in Cloud technology can be solved. This model also provides an access control technique in which access to data based on user privileges. Encryption maintains confidentiality, while authentication ensures only legitimate user accesses the data. Authentic users first encrypt their data and then stores into the cloud. Main purpose of this application is to use a new type of key called "Group Key" that represents a group of users. File Encryption Scheme will be enhanced to use group keys. The proposed framework is extensible enough that it allows an organization to decrypt the encrypted file.**

*Keywords* - **Access control, Cloud Computing, Cryptography, Hybrid Cloud, Private Cloud, Public Cloud, Security.**

## 1. INTRODUCTION

The cloud is not simply the latest fashionable term for the Internet. Though the Internet is a necessary foundation for the cloud, the cloud is something more than the Internet. The cloud is where you go to use technology when you need it, for as long as you need it, and not a minute more. You do not install anything on your desktop, and you do not pay for the technology when you are not using it. Cloud computing, where applications and files are hosted on a "cloud" consisting of thousands of computers and servers, all linked together and accessible via the Internet. Hence, you can access all your programs and documents from any computer that's connected to the Internet. It enables cloud customers to remotely store their data into the cloud so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources [1].

Successful examples of Cloud Storage Service Providers are Dropbox [2],Amazon's EC2 and S3 [3], iCloud [4], Nirvanix [5] etc. which provide data storage service in the pay-as-you use fashion at relatively low prices. For example, Amazon's S3 data storage service just charges $0.12 to $0.15 per gigabyte month. As compared to building their own infrastructures, users are able to save their investments significantly by migrating businesses into the cloud.

The benefits brought by this new computing model include but are not limited to: relief of the burden for storage management, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc [6]. With the prevalence of cloud services, more and more sensitive information are being centralized into the cloud servers, such as emails, private videos and photos, company finance data, government documents, etc [7].

On the surface, cloud storage has several advantages over traditional data storage. For example, if you store your data on a cloud storage system, you'll be able to get to that data from any location that has Internet access. You wouldn't need to carry around a physical storage device or use the same computer to save and retrieve your information.

As data owners store their data on external servers, there have been increasing demands and concerns for data confidentiality, authentication and access control [6]. Data is biggest asset to an organization & how confidentiality, authentication and access control can be outsourced. There is a threat to data owner that if CSP is malicious or CSP has some vulnerability. Hence data owner must have some way of ensuring the data is confidential from CSP.

I propose a framework that solves this problem – maintaining confidentiality of Data from CSP, using concepts of Cryptography (Encryption/ decryption, Symmetric & Asymmetric encryption), Hybrid Cloud, Existing organizational resources (Active directory).

The remainder of the paper is organized as follows. Section 2 presents proposed framework. Section 3

gives results. Section 4 concludes the paper and also suggests further extensions.

## 2. PROPOSED FRAMEWORK

Storing data on cloud can lead to leakage of sensitive information due to application and cloud vulnerabilities. In the proposed work, the concept of hybrid cloud infrastructure (i.e. combination of public and private cloud) is used with cryptographic approach in order to provide a secure and robust solution for securing the files on cloud yet maintains the confidentiality of data from CSP and auditing the employees. Confidential files can be encrypted and uploaded on cloud from managed client. A managed client is a computer or a device that has software required to encrypt the file and communicate with internal cloud. This would provide an Encryption solution to protect files stored on cloud and allow users to view/edit Encrypted files stored on cloud.

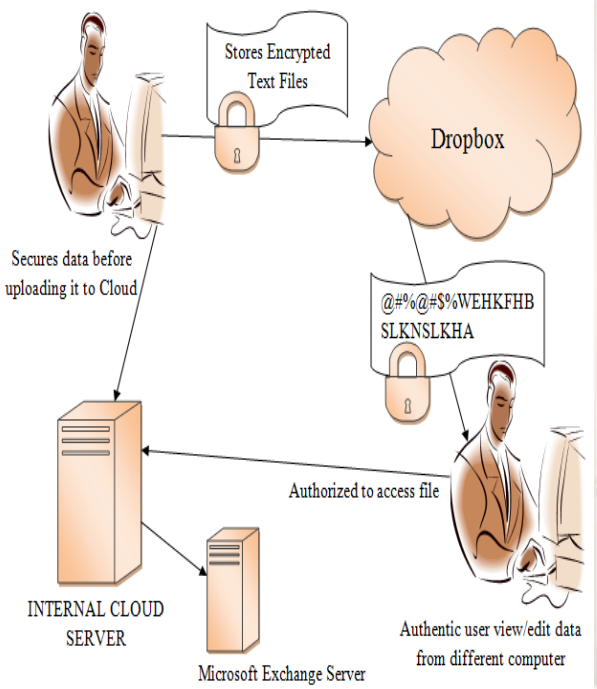


Figure 1 Design of proposed framework

Public Cloud will be only hosting the documents. It will only acts as Cloud as a Storage. Here, Dropbox has been used for this purpose (one can also use any other cloud service that offers Storage as a service). Dropbox provides a cloud based service to enable users to store and share files and folders with others across the Internet using file synchronization [2]. Each user stores their files in the dropbox (into the shared folder).

Private Cloud (Internal Cloud) will be used for the authentication, key management and access control. It uses database for maintaining information about registered groups and users, storing keys of users / groups, message authentication code of the files & the usernames and group names along with their different privileges. Hibernate [8] and Postgresql [9] has been used for this purpose (one can also use any other RDBMS). Hibernate is an object relational mapping (ORM) library for the Java language, providing a framework for mapping an object-oriented domain model to a traditional relational database. Postgresql is an object-relational database management system.

The given framework leverages on existing resources for authentication purposes. Every organization maintains information regarding their users in various forms. One of the widely used repository for maintain this information is Microsoft Active Directory [10]. Active Directory is a directory service created by Microsoft for Windows domain networks. It is based on the LDAP protocol for authentication purposes. It is a database that keeps track of all the user accounts and passwords in your organization. It also provides the facility of groups. Active Directory Groups contain users who are called members of the group. All permissions, authorizations and restrictions placed on the group apply to all the members of the group. For authentication purpose, a user credentials (i.e. username and password) has been used.

Encrypting your data before it is sent to the service provider ensures that if the provider's security measures are breached, your data is still secure. If someone does get your data, they need the proper credentials for viewing the files or all they get is gibberish.

Initially, Users will enroll by using their existing LDAP login credentials. After a server has been specified and authenticated, users can enroll by using their existing LDAP login credentials and get their keys from the Internal Cloud Server. The end-user will be presented with a login screen where they can enter their LDAP login credentials. Upon authentication, the user's keys will be downloaded from the server and onto the device. Users can then begin to encrypt / decrypt content.

According to fig.1, Data Owner (creator of file) first encrypts the file and then stores into the cloud. Typically file encryption mechanism includes metadata attached to the protected object that contains information about how to decrypt the protected object. This metadata is part of the encrypted file header and is always inserted at the beginning of the file. This metadata allows individual users to access the file.

Two keys used for encrypting a file:

1. File Data encrypted to unique File Encryption Key (FEK). This is also referred as Symmetric Key.
2. FEK encrypted to each user's public key. This is also referred as Encrypted Symmetric Keys (ESK).

If numbers of users who access encrypted file grow, associated metadata grows as well. This increases the encryption overhead significantly if original files size is small. Additionally, as new users joins and leaves the group, the management of metadata associate with each file causes significant maintenance issues with no easy way to manage it. And the only way to manage it is to re-encrypt the file which is costlier operation.

A group is basically a distribution list whose members will be resolved first in order to retrieve the individual keys for every group member. File Encrypting application then uses the member keys to protect the encrypted object. However, changes to the group membership require changes to the protected objects. On the other hand, this group feature does not require an infrastructure for re-encryption and thus is immune to all sorts of Group Membership changes.

In a typical organization users are grouped into groups. A Group Key is a key that is used by all users in a group. This feature is intended to be used in enterprise environments which host there data on third party cloud for the protection of shared files. Users intended to use the Group Key must be present in Active Directory and there keys reside on Internal Cloud. The Group Key, as an Internal Cloud managed key. Users will get the key when they require it and immediately after use key will be flushed.

At the time of login, each user is given its' private key and also notifies with the groups to which user belongs, by the server.

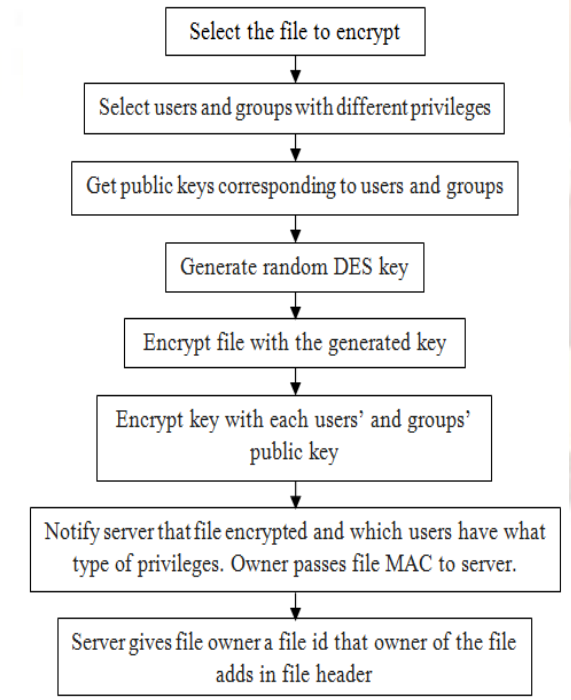


Figure 2 Steps of Encryption

Fig. 2 represents the steps for encryption. Once data owner selects a file to encrypt along with the users and groups with different privileges (read/write) with whom user (owner of the file) wants to share its' file. Then the user will get the public keys corresponding to selected users and groups from the internal cloud server. The application generates a random DES key used for the encryption of the file. The FEK (random generated key) used for encrypting the original file encrypts with each users' and groups' public key (only the authorized ones who have access to the file) and the resulted encrypted keys have been appended to the header of the encrypted file with their usernames and group names. The encrypted file header (metadata) contains the file id (generated by the server), an integer value that represents the number of users and groups with whom the file has been shared, usernames & group names and their corresponding ESK (key used for the decryption of the encrypted file). Once the file is encrypted, owner notifies the server and also which users and groups have what type of privileges. Owner also sends the Message Authentication Code (MAC) of the content of the file to the Internal Cloud Server. The usernames or group names along with their access privileges and the MAC of the file has been stored by the service provider in its database for future use. Then the server returns a file id (generated by the server) to the file owner that he/she adds it to the file header.

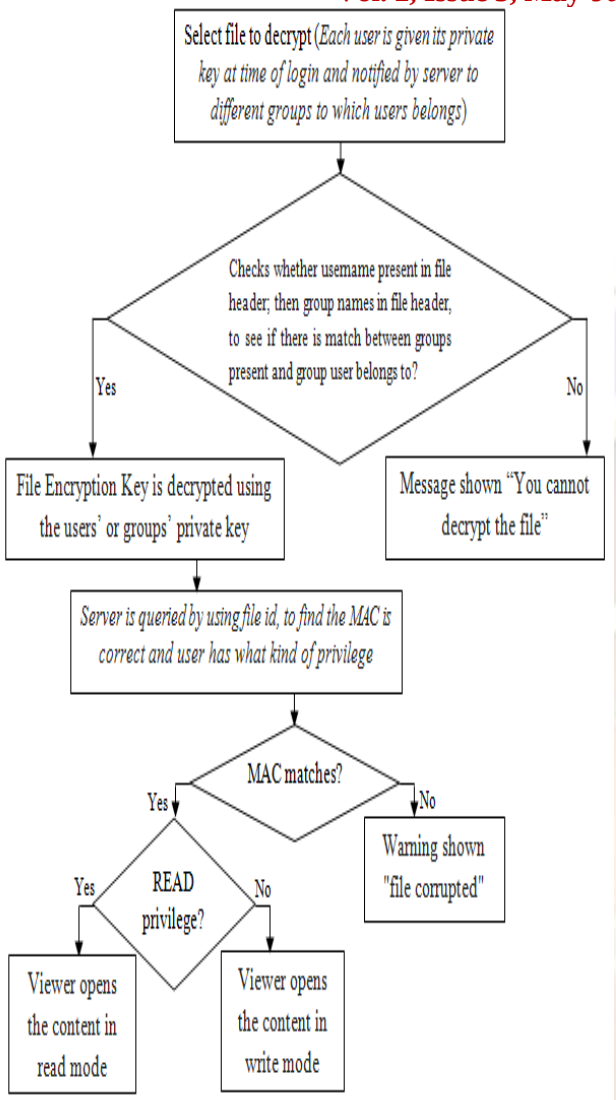


Figure 3 Steps of Decryption

Fig. 3 represents the steps of decryption. User selects a file to read or edit from the shared folder of its dropbox account. Now, the application running at clients' side tries to decrypt it. The running application first attempts to check whether the login username present in encrypted file header or not, then group names in file header, to see if there is a match between groups present and group user belongs to (if login user is part of some group and there is a match between groups present and group user belongs to, then user will get the private key of that group at the time of its use, once user gets the key, use it and then the application automatically flush it from the memory). If yes, the symmetric key (also known as File Encrypted Key, FEK) will be decrypted by using the corresponding private key else display a message "you cannot decrypt the file". As the symmetric key used for the encryption of the file has been decrypted by the application. Now, the encrypted data present in the file will be decrypted by the FEK. After the decryption, new message authentication code (MAC) of the content of the file has been generated. Then, the application running at clients' side queries internal cloud server using file id, to find whether the new generated MAC and saved MAC to the server, both are same or not, if MAC doesn't match then the "file corrupted" warning will be shown. If MAC matches, then the server checks in its' database that the login user has what kind of privilege and returns it to client. Now, the application checks if user has read privilege then viewer opens the data in read mode otherwise in write mode. If the viewer opens in read mode then the login user will not be able to make changes (neither add nor delete the content) to the data. But in write mode, user can make changes (must have to press the save icon) and these changes will be reflect in future. The updated file has been again encrypted by the running application.

## 3. RESULTS

The implementation of the access control based data security application is done in java.

Table 1 Table showing the relationship between users and groups

| *User name* | *Belongs to group* |
|---|---|
| User1, user2 | Group1 |
| User3, user4 | Group2 |

**File access:**

Read users- user1, user3
Read Groups- group1
Write users- user2, user4
Write group- group2

Table 2 Table Showing the Users' Access Permission on File

| *Username* | *User access permissions on file* | *Reason* |
|---|---|---|
| User1 | Read | User1 and Group1 both have Read permission. |
| User2 | Write | User2 has Write and Group1 has Read permission. |
| User3 | Write | User3 has Read and Group2 has Write permission. |
| User4 | Write | User4 and Group2 both have Write permission. |
| User5 | No access | User5 has no access, also doesn't belong to any group. |

Table 3 Test results observed in different scenario

| *Test Results* | *Result Observed* | *Result Description* |
|---|---|---|

| | | |
|---|---|---|
| When user with access privilege tries to access the file. | User has access to the file. Read user has no right to make changes (neither add nor delete the content) to the data. But write user can make changes and these changes will reflect in future. | User has corresponding decryption key (symmetric key, decrypted by user's private key only if user is authorized) to decrypt the content of the file. |
| When a read user tries to write the file using some other editor. | The file will be detected as corrupted when used by other users in future. | The updated file can be uploaded by user but user has no right to update the MAC stored into the internal cloud. Now, when file will be used by other users then MAC varies hence file will be resulted as corrupted. |
| When illegitimate user tries to access the file. | User has no right to access the file. The file will not be decrypted by the user. | User has no corresponding decryption key as he doesn't have the private key to decrypt the symmetric key. |
| When illegitimate user tries to create a new file. | User cannot upload the MAC corresponding to the file id on internal cloud. | User can upload the file on cloud using malicious CSP but he is not authenticated to the internal cloud as he doesn't have the active directory credentials. |

## 4. CONCLUSION AND FUTURE WORK

At present, there has been a lot of attention on cloud data security. This model shows how authorized user can securely access the data stored by data owner. The combined approach of access control and cryptography is used to protect the data that now resides at cloud.

The following conclusion can be drawn from present study:

- In the proposed framework concept of group key has been introduced which is a step towards reducing the file metadata & it solves the problem of re-encrypting the file which arises in case group membership of user changes. But there is a lot of computation overhead to cloud servers hence some extensions can be used to reduce the overhead.
- The concept of groups reduces file size hence also reduces cost of uploading the encrypted file (which is the main feature of cloud computing).
- The proposed framework is extensible enough to support auditing of data on Cloud Service provider and it enables an organization capability to decrypt the encrypted file just in case a situation arises.

Some direction for further extension is as follows:

In the existing implementation only the files have been encrypted similarly folders can also be encrypted where each file in a folder is encrypted with a symmetric key(FEK), file symmetric key is encrypted by one more symmetric key that is at folder level and folder level symmetric key is encrypted by users public key (ESK).

## REFERENCES

[1] P. Mell and T. Grance, The NIST definition of cloud computing, *National Institute of Standards and Technology*, vol. 53, p. 50, 2009.

[2] Dropbox, Online at https://www.dropbox.com/.

[3] Amazon Web Services (AWS), Online at http://aws.amazon.com.

[4] iCloud, Online at https://www.icloud.com/.

[5] Nirvanix, Online at http://www.nirvanix.com/.

[6] A. Fox and R. Griffith, Above the clouds: A Berkeley view of cloud computing, *Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Tech. Rep. UCB/EECS, vol. 28*, Feb 2009.

[7] S. Kamara and K. Lauter, Cryptographic cloud storage, in *Proceedings of Financial Cryptography: Workshop on Real-Life Cryptographic Protocols and Standardization 2010*, January 2010, pp. 136-149.

[8] Hibernate, Online at http://www.hibernate.org/.

[9] PostgreSQL, Online at http://www.postgresql.org/.

[10] Active Directory, Online at http://en.wikipedia.org/wiki/Active_Directory.