IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-3, Issue-5 E-ISSN: 2347-2693

Cryptographic File System for Secured Group Communication

Blessy Paul V [1*] and Ms. Jibi K George [2]

[1*,2]Department of Computer Science, Nehru College of Engineering and Research Centre, Pampady , Kerala, India

www.ijcseonline.org

Received: May /02/2015 Revised: May/09/2015 Accepted: May/22/2015 Published: May/30/ 2015

Abstract— Cloud computing is the delivery of computing and storage capacity as a service to users. The cloud has huge potential when it comes to storing, sharing and exchanging files, but the security provided by cloud services is questionable. Users, after uploading their files, have no control anymore about the way their data is handled and the location where it is stored. Considering both corporate and personal data which is often secret and sensitive in nature, one should not blindly entrust it to a cloud storage provider. In order to ensure the confidentiality of date during transferring it to the cloud storage, we need some encryption techniques. In this paper, we present a cryptographic file system based on Multiple-Key Public-Key Cryptography which is designed for enhancing the cloud data storage security for an organization. Here, files are encrypted by the administrator of the organization before they are uploaded to the cloud storage providers. Therefore, the cloud storage providers can't access the users' data. This encryption technique also allows the sharing files within a group of users of an organization. A key feature is that it handles changes in group membership and modification of files in an extremely efficient manner, by using the same encryption method.

Keywords—*Dynamic Membership, Multiple Key public key Cryptography, Cloud Storage*

1. INTRODUCTION

The tendency of modern companies and organizations is that they outsource their storage systems into the cloud. Storing data in the cloud has various advantages: cloud based storage can be distributed and redundant, providing better dependability and it is much cheaper for a company to outsource its data than to build and maintain its own data warehouse. In addition, a cloud storage provider solves the backup and the off-site backup.

The big question is trustworthiness and privacy. The problem is that a cloud storage provider also has access to the stored data, because usually the authentication and authorization is also done by this provider, think about e.g. Google Storage, a user has to log in by providing a user name and a password in order to access his or her files. After the authentication, user authorization is based on the information in an Access Control List (ACL), a simple list of users and their permissions. The problem with ACL based protection is that it needs an entity, which enforces the access control policy defined by the ACL by blocking the access to a non-permitted file.

This is a simple and fast approach, however, problematic. Firstly, if the cloud storage provider is compromised, an attacker can access every file in contempt of the ACL. Secondly, the administrators of the cloud storage provider can override the ACL settings, so they have access to the users' private files. Although ACL based systems may be implemented in more complex and secure way, the basic idea is still the same.

In cryptographic file systems there is no problem with an outside attacker or the curiosity of the cloud administrators, because every file is encrypted before being uploaded to the storage cloud. On the other hand, because of encryption, sharing files becomes problematic.

This paper presents, a cryptographic file system based on Multiple-Key Public-Key Cryptography Encryption techniques that builds on existing traditional storage clouds. It designed in such a way that it is extremely flexible, supports dynamic groups and handles changes very efficiently. This means that it is possible to define traditional permissions on the file system in a simple way, so that it is easy to understand and implement. Security is achieved by the encryption of files before uploading them to the cloud by the administrator of the organization.

Simplicity is achieved with an ACL like abstraction towards users which is easy to adopt and use. It also supports groups and content sharing within a group, which makes collaborative work easier. Collaborative applications can leverage cloud services greatly, but they need to be secured. For flexibility, encrypting the content with different keys so that a subset of the content can be shared at any time.

2. LITERATURE REVIEW

2.1. Cloud Data Storage

As a relatively new business model in the computing world, cloud computing is defined as a model for enabling

ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

In recent years, this innovative computing technology has drawn much attention in the fields of industry and academy. The great flexibility and economic saving of cloud computing are motivating all kinds of users, such as customers, enterprises, and even government organizations, to adopt cloud.

Cloud data storage consists of a huge number of storage devices that distributed throughout the network; it also provides users with the normal structure of the cloud data storage, which include distributed file system, resource pool, service interfaces and service level agreements (SLAs), etc. Cloud storage is a branch of Cloud Infrastructure as a Service (IaaS) in cloud computing and it is working to provide the data, and reduce infrastructure costs by storing data remotely. Cloud data storage works to provide storage service for different levels of customers as the cost of storage depends on the space required the ability and bandwidth. To manage the contents of the data stored in the cloud data storage can rely on Service Oriented Architecture (SOA), Web Service (WS).

2.2. Security Issues in Cloud Storage

Secure Storage means to protect data from unauthorized access. From literature review on the secure storage always the risk is come from inside or the threat from outside In fact, Main risks that make companies do not tend to cloud computing is the secure storage of data. Cloud computing is an emerging paradigm, but its security and privacy risks has been attracting significant attentions of cloud users and cloud providers. One of the important reasons is that cloud users have to trust the security mechanisms and configuration of the cloud provider and the cloud provider itself. In the community of industry and academy, cryptographic technique is currently treated as one of the key techniques to solve security and privacy problems existed in cloud computing environment. The main concept for secure storage in the cloud computing is to encrypt data in a reliable environment before being sent out of the cloud in an environment that is unreliable.

2.3. Encryption on Cloud Data

The cloud has huge potential when it comes to storing, sharing and exchanging files, but the security provided by cloud services is questionable. Users, after uploading their files, have no control anymore about the way their data is handled and the location where it is stored. Even worse, users have no means to control access to their data. Considering both corporate and personal data which is often secret and sensitive in nature, one should not blindly entrust it to a cloud storage provider.

The main challenge of cloud storage nowadays is to guarantee integrity, confidentiality and control over all stored data. Users rarely have the possibility to check whether a given cloud provider satisfies these criteria. Furthermore, the quick spread of cloud storage solutions and their user-friendliness can possibly undermine the awareness of users regarding the transmission and storage of their confidential data. Therefore, users should apply security-oriented cloud storage middleware that, on the one hand, keeps the usage of the cloud simple and fast, and, on the other hand, enforces the application of strong cryptographic algorithms and protocols in order to keep private and confidential data from being leaked.

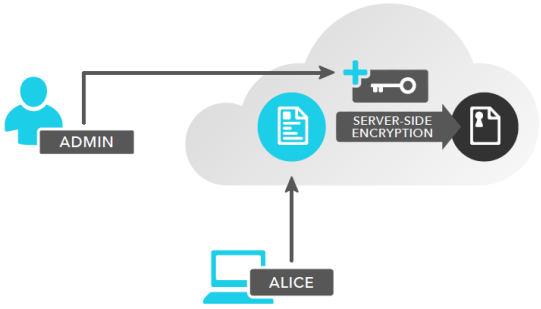

Figure2.3. Using Traditional Cloud Storage Middleware

In other words, the user has to trust the cloud storage provider for being honest and not revealing her private data. Without the blind trust in the cloud storage provider, the user could choose to encrypt her files one-by-one and only upload these secure files to the cloud. This works well for things like backup, but only as long as the user does not want to share her files. As the amount of data and the number of people involved in the sharing rises, this approach becomes intractable. The overhead with group management, invitation of new group members, and revocation of expired permissions quickly becomes a burden. This approach is just not flexible enough for collaborative use.

2.4. Administrator Level Encryption

Every file is encrypted before being uploaded to the server.This encryption is carried out by the administrator of the organization. In these systems there is no problem with hackers or the curiosity of the administrators of the cloud, because every file is encrypted. On the other hand, because

of the encryption, sharing is problematic. Alice would rather encrypt her files by the admin before uploading them to cloud. Using a strong encryption ensures her that no one can read the content without the encryption key.

Nowadays tons of cryptographic tools are available for the public. However, none of them really support sharing in a dynamically changing environment so, Alice might be in trouble if she wanted to work with Bob using Google Docs, because only Alice can decrypt the file's content.

By using a simple storage cloud , Alice can have private files in the cloud. Bob and Carol cannot read Alice's private files, because the storage cloud would block their access. Alice, by using simple encryption on her private files can prevent unauthorized access. Alice cannot share her files with Bob easily because her files are encrypted with a one-key encryption which is done by the admin of organization. On the other hand, by using this cryptographic file system based on multiple-key public-key cryptography encryption, Bob and Carol can easily share their files together. With this security solution, a group of users of an orgabization can share secure content by the control of the admin of organization without trusting any cloud provider.

3. PROBLEM DESCRIPTION

3.1. Overview

The endpoints of the system are the users in an organization with an admin and their software clients who would like to share files with each other. These users form groups with those users whom they would like to collaborate within the organization. A group consists of a set of users or even other groups. Every group of users has its content in a database of cloud. This contains all the files of the group in an encrypted form. This encryption is carried out by the administrator of the organization before uploading the data into cloud.

First assumption about the entities is, the users of the system would not always like to collaborate with the same people. This means that the aforementioned groups dynamically change in time. Users can be added or removed from an existing group; new groups may appear while others disappear. During the lifetime of a group the content shared would also change. Files and directories can be added, removed or changed. Currently, in storage clouds the service providers enforce the security of the cloud objects using ACL's, which are vulnerable to software bugs or override by an administrator or hacker. The service providers are assumed to be honest but curious; therefore, the service providers cannot be fully trusted.

3.2. Requirements

First list the security requirements. The first objective is secrecy: Any user not part of the group cannot access any content in the group's database. To further characterize the secrecy of the group it needs to specify what previous users and future users of a group can access. Any previous user of a group who has knowledge of group keys until his or her removal from the group should not be able to read any new content created in the group's database since his/her departure from the group. This is known as Forward Secrecy.

Any new user that joins the group only knows the group keys from the moment of entering into the group. The new user should be able to read all contents currently in the group's database but not deleted files. This we will refer to as Weak Backward Secrecy. Simply expecting Backward Secrecy would mean that the new user would only be able to access new content that is created during his/her participation in the group.

Imagine that a a group of users have agreed in some private key that is proposed by the admin of organization and admin encrypts every file in the group and users decrypt the corresponding files by using their private keys. This solution at first glance seems to satisfy everything that would expect from such an environment, but very quickly it discovers that there are many problems with this construction. The main problem is that if a user is removed from the group then the private keys would have to change, this would mean that all the existing contents of the group's database would have to be re-encrypted. In order to solve the latter problem efficiently, this paper proposes an efficient Multiple-Key Public-Key Cryptography based encryption method that derives keys from the modular functions to encrypt as well as decrypt the files in the group's database.

4. CRYPTOGRAPHIC FILE SYSTEM

4.1. Organizational Over View

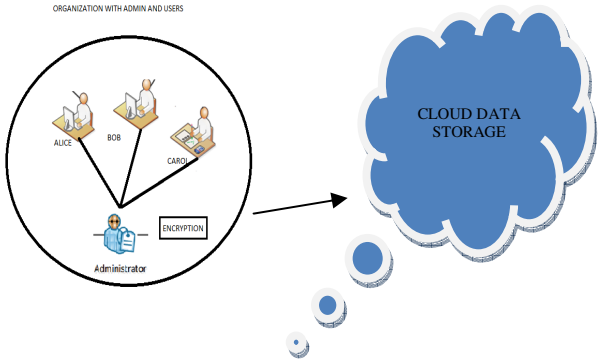

Figure4.1. Organization Overview of Encryption

A private cloud which is allocated for a particular organization with an administrator and 'M' number of users. Here each user wants to store their own private data files and also the data files which are shared between the users in the private cloud allocated for the organization. So before uploading the files into the cloud for the storage purpose, an efficient encryption should be carried out on it, which ensures the security of the uploaded data.

Here the encryption is performed by the administrator of the organization, so that the cloud provider can't read the encrypted files in the cloud. Also it provides the security from outside hackers. So if anyone who is not part of the organization can simply retrieve only the encrypted files from cloud, where the outsider is unaware about the decryption keys since it is the private keys of each user.

4.2. Multiple-Key Public-Key Cryptography

The method using for encryption and decryption is based on Multiple-Key Public-Key Cryptography. Here each user transfers their private files and shared files with other users to the administrator of the organization. At the admin level encryption of the files will be carried out based on the Multiple-Key Public-Key encryption method. For which it uses an encryption key along with a modular function. Before that administrator will assign some private keys to each user in the organization. Modular function consisting of private keys which have assigned to users and the encryption key used for encrypting the data. Encrypted files are then stored in cloud storage. Based on the modular function only the intended users can decrypt the files from cloud. Consider there are 5 users in the organization: Alice, Bob, Carol, Dave and Ellen

That is Alice's private file is encrypted by the admin in such a way that the only Alice can decrypt it from the cloud by using her private key. Likewise the modular function will be created. In Similar way if Alice and Bob sharing a particular file, first they will transfer it to the admin. Admin will generate an encryption key and modular function based on their corresponding private keys. So that only Bob and Alice can decrypt it from cloud. Administrator will assign private keys to each user.

USER	PRIVATE KEY
Alice	X_1
Bob	X_2
Carol	X_3
Dave	X_4
Ellen	X_5

Table4.1. Users and their Private Keys

Admin will create the modular function as X_i Mod N=K. And encrypting the private files based on this function. So only the corresponding user can decrypt it. And in the case of shared files admin will create the function and encryption key 'N' based on their private keys. So all users in the shared group can retrieve it.
For example, Suppose Alice, Bob, Carol, Dave and Ellen are assigned with the private keys as shown in below table

USER	PRIVATE KEY
Alice	2730
Bob	13090
Carol	8778
Dave	7590
Ellen	33495

Table4.2. Private Keys for Each User

The function created for each user's private files are as

Alice → X_1 Mod 13=0
Bob → X_2 Mod 17=0
Carol → X_3 Mod 19=0
Dave → X_4 Mod 23=0
Ellen → X_5 Mod 29=0

And likewise if Alice and Bob sharing a particular file, it can be encrypted by using the function X_i Mod 70=0.So only Alice and bob have the private keys satisfying this condition. So only they can decrypt it.

In similar if Alice, Bob, Dave, And Ellen sharing a particular file, it can be encrypted by using the function X_i Mod 5=0 .Likewise we can create encryption function for any user groups.

4.3. Group Modification

Next the considerable change will occur in the organization if a particular member is leaving from the group. Because the future files can't be decrypted by him after he has removed from the group. To meet this condition the admin will reconstruct the modular function for each set of file's encryption.

For an example suppose Alice has removed from the organization, then the file which was shared among Alice, Bob, Dave and Ellen will be re-encrypted by using the new modular function Xi Mod 55=0. So Alice can't be decrypted it in the future.Similar changes have to be made if a new user is joining in the organization. For that the private keys and the modular functions are created already for L number of users. And M be the number of users currently joined in the organization, where L>M. If the number of users in organization is exceeding the number L, the admin will reconstruct the private keys. And thus the modular functions also re-constructed. So re- encryption again takes place. For the reliability admin will assigns private keys for a very large number of L.

5. ANALYSIS

5.1. Security

5.1. 1) Forward Secrecy:

Algorithm guarantees that only those users can read the contents of Database who have the private keys assigned by the admin. This means that a user 'U' will not be able to read any files after the time of his/her departure from the group. All modified files will have new encryption key and encryption function, which means that the Forward Secrecy constraint will hold and no former member will be able to decrypt any of the new files.

5.1.2) Weak Backward Secrecy:

If a user U has been added to the organization, already assigned new private key will allocate to him by the admin. Based on which new functions are created and encryption is performed. This means that user u will only be able to access those files that are in the database during his or her stay in the group, satisfying Weak Backward Secrecy.

5.2 Advantages

End-to-end Encrypted**:** Encryption and decryption are done on the client side (i.e. at organization). No entity is able to recover the data, except for the owner or admin of organization herself and users authorized by the owner. No trust in the cloud storage provider is required. Data stays as safe as if it was stored securely on your own system. Shareable: The owner can invite anyone with ease to collaborate and share files with. Only an e-mail address is required to send the invitation. Shared files and folders can be jointly modified, synchronization is performed automatically. Any number of files and directories can be shared among any number of users.

6. CONCLUSION

This paper, presents A Cryptographic File System for Enhancing the Cloud Security, based on Multiple-Key Public-Key cryptography. In which, files are encrypted before they are uploaded to the cloud storage providers, by the admin of the organization, therefore, not even the cloud storage providers can access the users' data.

A key feature is that it handles changes in group membership and modification of files in an extremely efficient manner. So if a new member is joining to the group or an already existing member is leaving from the group won't affect the security factors of user files. In terms of security, this cryptographic file system also achieves Forward Secrecy and Weak Backward Secrecy. In the case of user authentication and authorization is also handled by the administrator of organization in a very efficient way. So only the allowed members are participating in file encryption and decryption.

Another important feature is the ability to share content between groups with less complexity, a feature which has not yet been implemented in any of the cryptographic file systems. Even though all stored files are in encrypted form, while using this cryptographic method, it allows the sharing of file mechanisms between any numbers of users in the organization.

REFERENCES

[1] Istv´An L´Am., Szilveszter Szebeni., Levente Butty., (**2012**), "Tresorium: Cryptographic File System For Dynamic Groups Over Untrusted Cloud Storage", Proceedings of the 41st International Conference on Parallel Processing 14, pp.1594–1603.

[2] Peng Yong., Zhao Wei., Xie Feng., Dai Zhong-Hua., Gao Yang., Chen Dong-Qing.,(**2012**), "Secure Cloud Storage Based On Cryptographic Techniques" Proceedings of the IEEE INFOCOM, 14, pp.1594–1603.

[3] I. Lam., S. Szebeni., And L. Buttyan., (**2012**) "Invitation-Oriented Tgdh: Keymanagement For Dynamic Groups In An Asynchronous Communication Model," In Submitted To 4th International Workshop On Security In Cloud Computing, 15, 17, PP.1684–1695

[4] V. Sriram., G. Narayan,, and K. Gopinath,, (**2007**) "SAFIUS - A secure and accountable filesystem over untrusted storage," In Storage Workshop, 2007. SISW '07. Fourth International IEEE, pp. 34–45.

[5] D. Grolimund., L. Meisser., S. Schmid., and R. Wattenhofer.,(**2006**) "Cryptree:A folder tree structure for cryptographic file systems," In Proceedings of the 25th IEEE Symposium on Reliable Distributed Systems, pp. 189–198.