# Data Security in Cloud Storage using RBAC

Harshal Karande
*Department Of Computer Engineering , Siddhant College Of Engineering, Savitribai Phule University Pune, India*.

***Abstract*- A cloud storage system is a collection of storage servers , provides long term storage services over the web. Storing knowledge in an exceedingly third party cloud system causes serious concern over knowledge confidentiality. A Secure cloud is a trustworthy source of information. Protecting the cloud is a very important task for cloud storage service providers. Now days, there is the need of low-maintenance system which automates administration regularly and also need of access control over network so that data security is maintained and insured. Role-based access control (RBAC) method controls access to computer or network resources based on the roles given to individual users within an enterprise. Roles are defined according to job skill, authority, and responsibility. The system provides policy access control and assured deletion. Assured deletion aims to provide cloud client on option of truly destroying their data backups upon request.**

***Index Terms*- Role-Based Access Control, Role-Based Encryption, Time-Based Assured Detection, Key-Manager, Cloud Storage.**

## I. INTRODUCTION

Cloud storage system provides benefits which come to an existence model that enables individuals and enterprises to outsource the storage of data backups to remote cloud providers at a low cost. However, cloud clients must ensure security guarantees of their outsourced data backups. The increasing popularity of cloud storage is leading organizations to consider moving data out of their own data centers and into the Cloud. Cloud storage service providers may make multiple backup copies of file and distribute them over the cloud for reliability, and clients may do not know the number or even the existence of these backup copies. Security concerns become relevant as outsourcing the storage of possibly sensitive data to third parties. To protect outsourced data, a direct approach is to apply cryptographic encryption onto important data with a set of encryption keys, yet maintaining and protecting such encryption keys will create another security issue. One specific issue is that upon requests of deletion of files, cloud storage providers may not completely remove all file copies, and eventually have the data disclosed if the encryption keys are unexpectedly obtained, either by accidents or by malicious attacks. Therefore, we seek to achieve a major security goal called file assured deletion, meaning that files are reliably deleted and remain permanently unrecoverable and inaccessible by any party. Two security issues are concerned; first, to provide guarantees of access control, in which only authorized parties can access the outsourced data on the cloud. Keeping data permanently is undesirable, as data may be unexpectedly disclosed in the future due to malicious attacks on the cloud or careless management of cloud operators. Second, it is important to provide guarantees of assured deletion, meaning that outsourced data is permanently inaccessible to anybody (including the data owner) upon requests of deletion of data. Access control is the action of controlling access over data in an internet world. It provides security. In computer security, general access control includes authorization, authentication, access approval, and audit. In this, system makes a decision to grant or reject an access request from an already authenticated subject, based on what the subject is authorized to access. Authentication and access control are often combined into a single operation, so that access is approved based on successful authentication, or based on an anonymous access token. The intent of Assured deletion is to produce cloud clients an option of truly destroying their data backups upon requests. On the other hand, cloud providers may replicate multiple copies of data over the cloud infrastructure for fault-tolerance reasons. Cloud clients do not know how many copies of their data are on the cloud, or where these copies are located since cloud providers do not publicize their replication policies. It is unclear whether cloud providers can reliably remove all replicated copies when cloud clients issue requests of deletion for their outsourced data.

## II. RELATED WORK

There are many hierarchy access control schemes which have been based on hierarchical key management (HKM) schemes, and approaches using HKM schemes to enforce RBAC policies for data storage are discussed in which describes Management process of access control evolution with encryption on outsourced data. However, these solutions have many limitations, if there are a many data owners and users involved, the burden involved in managing the key infrastructure that can be very high indeed. Even when a users access permission is invalidating, all the keys known to this user are changed as well as all the public values related to these keys need to be changed. An approach for the key management is Hierarchical ID-based Encryption (HIBE).

However, in a HIBE scheme, the length of the user identity becomes longer with the growth in the depth of hierarchy. In Attribute Based Access Control model, access is provided based on attribute of user. ABE scheme was introduced in. In this approach, the size of user key is not constant, and the revocation of a user will result in a key update of all the other users of the same role. By combining attributes, a role can be formed, which provides multiple authorities described in Access control policies are provided in . Time based file assured deletion, is first introduced in which files can be deleted securely and remain inaccessible after a predefined duration.

## III. METHODS

### A. Role-Based Access Control

Role-based access control implements a better security solution for accessing data on cloud. Roles in RBAC system are generalized to access permissions, and all users are generalized

to appropriate roles and contain access permissions only through the roles to which they are allowed, or through hierarchical roles, roles get access permission. Within an organization, there may be number of data users and many types of permissions, whose role and accordingly access changes. Managing all access through roles gives benefit to constitution and it also simplifies the management. Specifically, role-based access control system has three essential elements; users, permissions and roles. A role is a leading level representation of access control. User corresponds to actual world users of the internet system. User authorization can be accomplished solely; assigning users to extant roles and assigning access authority for objects to roles. Permissions give a definition of the access users that can have objects in the system and roles gives a definition of the authorization of users within an organization. In RBAC, there is hierarchical system that means; a role can derive access permission from other roles. Following diagram shows relationship between users, roles and permissions.

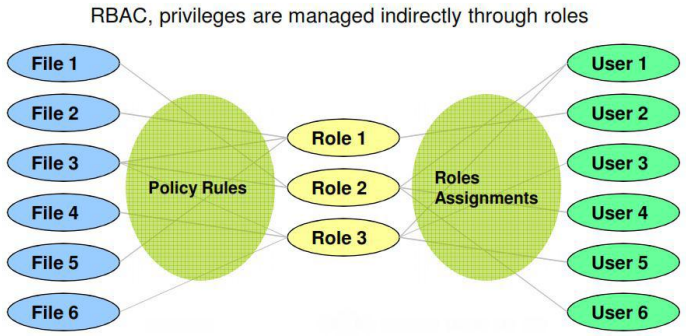


Fig. 1. Role Assignment to user

### B. Role-Based Encryption

In RBE scheme, the user who is the owner of the data encrypts the data in such a form that only authorized users can decrypt the data, users that possess definite access permission based on their role specified by role-based access control policies. Role allocates permission for accessing data according to their role and can also invalidate the permission from current user of role. Revoked user does not have any type of permission to access any encrypted data for the role.

Revocation of the user from the system does not impact on other users and roles in the system. In RBE, four basic types of entities are used; a system administrator which generates keys for users and allocates roles and provides authorization. A role manager gives access to user based on their role. Users are used to decrypt and access data files from cloud. Data files are stored on cloud by owner of the data.

### C. Time-Based File Assured Deletion

Data key, Access key and Control Key are the three types of cryptographic keys used to protect data files stored on the cloud. Data key is a key used to encrypt and decrypt data file via symmetric key encryption. Access key is used to maintain authorization of the user and control key is a key used to encrypt and decrypt the data key which is

maintained by ephemerizer. Time-based file assured deletion means that files can be securely deleted and remain permanently inaccessible after a predefined duration. One or more ephemerizer will be used, which advertise public keys with expiration dates. Data with a particular expiration date will be encrypted with the ephemeris public key with that expiration date. The straightforward approach would be to have each file encrypted with its own key K, and to store K, encrypted with the corresponding ephemerizers key, in the metadata of the file. However, this would require the file system to interact with the ephemerizer whenever each file was opened. It would also require a lot of storage in each files metadata, since although K would e a secret key. The file system will need to interact with the ephemerizer, upon reboot, to build a table of (symmetric) master keys, one for each possible expiration date, which the file system will keep in volatile storage. The file system generates a secret key Si, for each expiration time i, and all files with the same expiration time will be encrypted with the same Si. There is a one-to-one correspondence between file system secret master keys and ephemeral public keys kept by the ephemerizers. The metadata for a file will contain an indication of which S the file has been encrypted with. For instance, the metadata might contain the expiration time of the file.

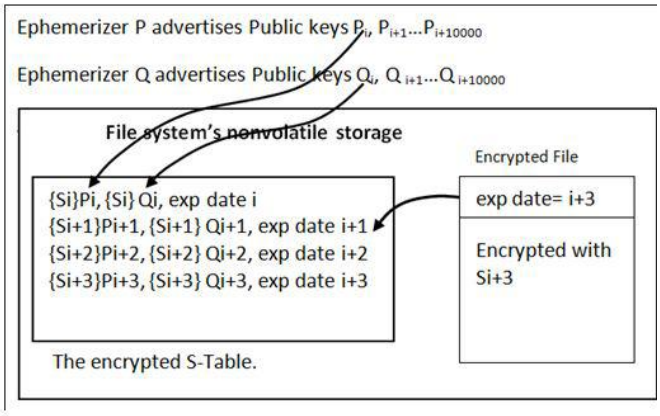


Fig. 2. Time-Based File Assured Deletion

## IV. IMPLEMENTATION DETAILS

### A. Problem Definition

Storing data in public cloud creates security issue. To overcome the data security issue, a key-manager module is being defined in the system which provides access to only those users who are authenticated. Attribute based access control provides access to users based on their attribute. However, in this approach, single user revocation results in key update of all other related user. To provide multiple authority and key management, role-based access control mechanism is used in the system. Role-Based Access Control is the provision of providing data access based on individual users role.

### B. Proposed System

In proposed system, role-based access control mechanism is used, in which data access permission is given to only those users who possesses appropriate role. User enters into the system. In the system, admin is work like role manager who give assign role to the user and accordingly role gets appropriate permission to access data on cloud. Permission sets are assigned to user according to his role. Key manager is used to provide access to only authenticated roles. Key manager performs various functionality such as managing access and by means providing more security to data access. In the system, role uses symmetric key functionality which provides same key for encrypting and decrypting file from cloud. Key manager uses asymmetric functionality which provides different keys for encryption and decryption function. So, a data provided by user is encrypted with symmetric key by role and transfer it to key manager in terms of data key and then key manager again performs encryption on data key by using asymmetric key and then stored it on cloud. If role wants to download the data from cloud, then he needs to provide authentication to key manager so that key manager decrypts data for him and then role can decrypt data and get original data. User authentication is done by using SPEKE algorithm which helps to maintain secure communication between key manager and user. We cannot store files permanently on cloud. Keeping data permanently is undesirable, as data may be unexpectedly disclosed in the future due to malicious attacks on the cloud or careless management of cloud operators. For deletion of files, time expiration policy is used which means files get deleted upon time expiration. The main security property of file assured deletion is that even if a cloud provider does not remove expired file copies from its storage, those files remain encrypted and unrecoverable.

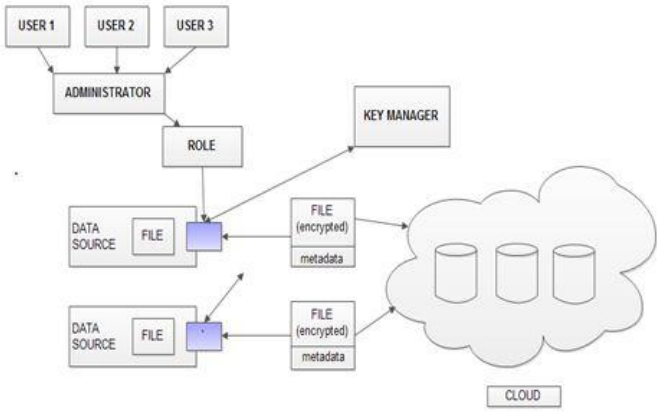


Fig. 3. Proposed System

C. Mathematical Model

Let U be users, R be roles, P be permission sets, which provides relationships between users, roles and permissions. Such that

U = {U1, U2, Un}
R = {R1, R2, Rn}
P = {P1, P2, Pn}

Here
I Represents the set of Input,
Let
I = { I1, I2, I3}
I1 = File Uploading;
I2 = File access structure;
I3= File Attributes;

And F is the set of function is provide functionality for Key Management
Let

F = {F1, F2, F3}
F1= AES Algorithm;
F2 = SPEKE algorithm;
F3= Role based access policy.

And O is the set of Outputs

O = {O1, O2, O3}
O1 = Key Generation(Symmetric);
O2 = Key Generation( master, public and secrete);
O3= File Decryption.

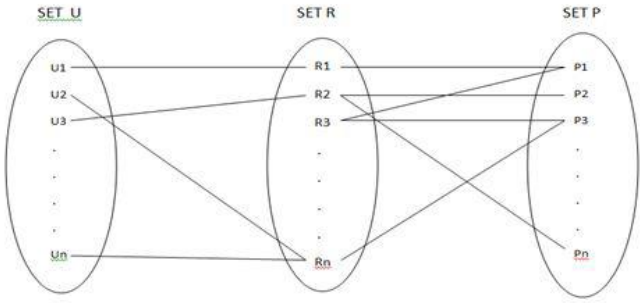


Fig. 4. Set Representation

In the below state transition diagram, U user, A admin who manages role for user according to their eligibility, Ris role which is assigned to user by admin, F1 is a function used to provide symmetric key to role, K is the key manager who manages access control for given role to provide more security by encrypting encrypted file provided by role, F2 is the function which provides asymmetric key pair, and C is the cloud where role/owner of the data is going to Upload/download files

D. In RBE System, following algorithms are used

Setup(): This algorithm takes some parameters and generates
master secrete key and public key.
Extract(): System Administrator(SA) execute this algorithm.
If user identity matched, then SA provides master key to the user, and then SA generates secrete key of user.

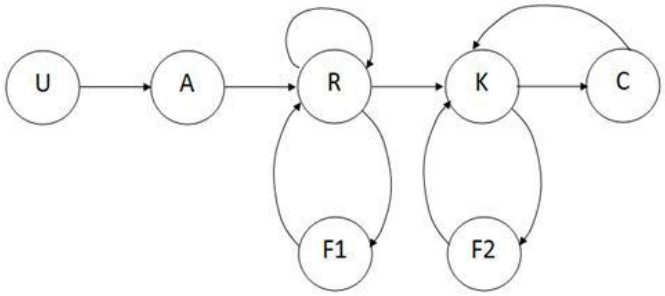


Fig. 5. State Transition Diagram

ManageRole(): SA execute this algorithm to manage role with identity from other role. Here, role hierarchy is maintained.
SA generates role public parameter and stored them on cloud.

AddUser(): Role administrator(RM) executes this algorithm in which RM gives role to user and also provides authentication. Role user list is updated in cloud.

RevokeUser(): RM execute this algorithm and sends user ID to the cloud, then cloud computes some parameters and send them back to RM from which RM replaces old role parameters with new parameters.
Encrypt(): Encryption is done by owner of the data and it stores cipher text of message to the cloud. This algorithm takes public key as input parameter and generates ciphertext and key K where K is used to encrypt original message.

Decrypt(): This algorithm is executed by those user who possess access according to their role. This algorithm takes public key, decryption key and ciphertext as input parameters and generates output by decrypting original message by using K.

E.AES Algorithm

Step1: SubBytes:
This step carried out byte-by-byte substitution during the process of forwarding. And for decryption InvSubBytes step used.This step consists of using a 16 16 lookup table to find a replacement byte for a given byte inan array.The entries are created in the lookup table.

Step 2: ShiftRows:
This step shifts the rows of the state array during the process of forwarding. Andfor decryption InvShiftRows step is used for Inverse Shift- Row Transformation.The goal of this transformation is to scramble the byte order inside each 128-bit block.

Step 3: MixColumns:
This step is used to mix up the bytes in each column separately during the process of forwarding. For decryption InvMixColumns step is used and stands for inverse mix column transformation. The goal is to further scramble up the 128-bit input block.The shift-rows step along with the mix-column step causes each bit of the ciphertext to depend on every bit of the plaintext after 10 rounds of processing.

Step 4: AddRoundKey:
This step is used to add the round key to the output of the previous step during the process of forwarding. The corresponding step during decryption is denoted InvAddRoundKey for inverse add round key transformation.

F. SPEKE Algorithm

SPEKE is a password-authenticated key agreement which establishes interaction between two parties based on their password knowledge. The system is maintaining role hierarchy, and secure communication between these roles is maintained by SPEKE. SPEKE prevents man in the middle attack by providing password authentication process. The simple password exponential key exchange (SPEKE) has two stapes.

Step 1: In the first step, shared key is being established, but instead of the commonly used fixed primitive base, a function converts the password into a base for exponentiation. Then users start out to choose two random numbers. Step 2: In the second step of SPEKE, both users confirm each others knowledge via shared key before proceeding touse it as a session key.

## V. EXPERIMENTAL RESULTS

In the experiment, we are calculating values of time taken for encryption and decryption of file.

| Sr. No | Key Size | Data Size | Encryption Time in ms | Decryption time in ms |
|---|---|---|---|---|
| 1 | 256 | 1024 Bits | 2435038 | 242439 |
| 2 | 256 | 2048 Bits | 2790355 | 4141310 |
| 3 | 256 | 3072 Bits | 7237362 | 8568297 |
| 4 | 256 | 4112 Bits | 8921497 | 3836031 |
| 5 | 256 | 5140 Bits | 7340987 | 8681938 |
| 6 | 256 | 6168 Bits | 8129473 | 9510442 |
| 7 | 256 | 7196 Bits | 6381975 | 7742940 |
| 8 | 256 | 8224 Bits | 3649988 | 4980934 |
| 9 | 256 | 9252 Bits | 6414279 | 7775244 |
| 10 | 256 | 10280 Bits | 7597358 | 9058393 |

TABLE I. ENCRYPTION AND DECRYPTION TIME FOR A GIVEN DATA SIZE

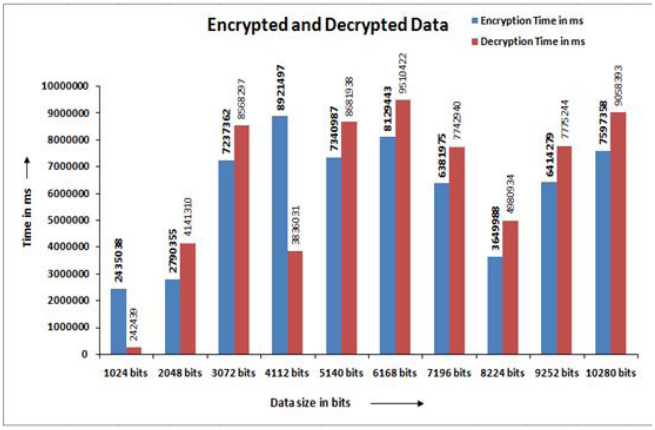


Fig. 6. Performance in Time

## VI. CONCLUSION

In this paper we try to provide data security by maintaining different key manager who manages encrypted key over encrypted data. RBAC contain some privileges and access policies Based upon authorization and access permission policies, user can access data from cloud. An active file on the cloud is encrypted with a data key, which can only be decrypted by the key manager. A file becomes deleted when its associated
policy is revoked.

## REFERENCES

[1] Yang Tang, Patrick P.C. Lee, John C.S. Lui, Radia Perlman, Secure
Overlay Cloud Storage With Access Control And Assured Deletion,
IEEE Transactions On Dependable And Secure Computing, vol. 9, no.
6, November/December 2012.

[2] Lan Zhou, Vijay Varadharajan and Michael Hitchens Achieving Secure
Role-Based Access Control on Encrypted Data in Cloud Storage, IEEE
Transactions on Information Forensics and Security, Vol. 8, No. 12,
December 2013.

[3] Vijay Varadharajan and Michael Hitchens Design and specification of
role-based access control policies, IEEE Transactions on Information
Forensics and Security, Vol. 147, No. 4, August 2002.

[4] Zahir Tari and Shun-Wu Chan Role-based access control for intranet
security, IEEE Internet Computing, Vol. 1, No. 4, September 1997.

[5] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski,
G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, A View of
Cloud Computing. Comm. ACM, vol. 53, no. 4, pp. 50-58, Apr. 2010.

[6] R. Perlman, File System Design with Assured Delete, Proc. Network
and Distributed System Security Symp. ISOC (NDSS), 2007.

[7] Y. Tang, P.P.C. Lee, J.C.S. Lui, and R. Perlman, FADE: Secure Overlay
Cloud Storage with File Assured Deletion, Proc. Sixth Intl ICST
Conf.Security and Privacy in Comm. Networks (SecureComm), 2010.

[8] H. R. Hassen, A. Bouabdallah, H. Bettahar, and Y. Challal, Key
management for content access control in a hierarchy, Comput. Netw.,
vol. 51, no. 11, pp. 31973219, 2007.

[9] S. D. C. Di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P.

Samarati, Over-encryption: Management of access control evolution on
outsourced data, in Proc. VLDB, Sep. 2007, pp. 123134.
[10] C. Blundo, S. Cimato, S. D. C. Di Vimercati, A. D. Santis, S. Foresti,
S. Paraboschi, et al., Efficient key management for enforcing access
control in outsourced scenarios, in SEC (IFIP), vol. 297. New York,
NY, USA: Springer-Verlag, May 2009, pp. 364375.