ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Decentralized Access Control with Anonymous Authentication of Data Stored in Clouds

Nampally Sushma
M.Tech Student,
Department of CSE,
AMR Institute of Technology,
Mavala, T.S, India.

Mrs.B.Manasa
Associate Professor
Department of CSE,
AMR Institute of Technology,
Mavala, T.S, India.

Abstract: ***Cloud computing's multi-tenancy feature, which provides privacy, security and access control challenges, because of sharing of physical resources among untrusted tenants. In order to achieve safe storage, policy based file access control, policy based file assured deletion and policy based renewal of a file stored in a cloud environment, a suitable encryption technique with key management should be applied before outsourcing the data. In this paper we implemented secure cloud storage by providing access to the files with the policy based file access using Attribute Based Encryption (ABE) scheme with RSA key public-private key combination. Private Key is the combination of the user's credentials. So that high security will be achieved. Time based file Revocation scheme is used for file assured deletion. When the time limit of the file expired, the file will be automatically revoked and cannot be accessible to anyone in future. Manual Revocation also supported. Policy based file renewal is proposed. The Renewal can be done by providing the new key to the existing file, will remains the file until the new time limit reaches.***

Keywords: ***cloud storage, renewal policy, decentralized access, policy based access.***

1. INTRODUCTION

Now a days cloud computing is a rationally developed technology to store data from more than one client. Cloud computing is an environment that enables users to remotely store their data. Remote backup system is the advanced concept which reduces the cost for implementing more memory in an organization. It helps enterprises and government agencies reduce their financial overhead of data management. They can archive their data backups remotely to third party cloud storage providers rather than maintain data centers on their own. An individual or an organization may not require purchasing the needed storage devices. Instead they can store their data backups to the cloud and archive their data to avoid any information loss in case of hardware / software failures. Even cloud storage is more flexible, how the security and privacy are available for the outsourced data becomes a serious concern. There are three objectives to be main issue Confidentiality – preserving authorized restrictions on information access and disclosure. The main threat accomplished when storing the data with the cloud. Integrity – guarding against improper information modification or destruction. Availability – ensuring timely and reliable access to and use of information.

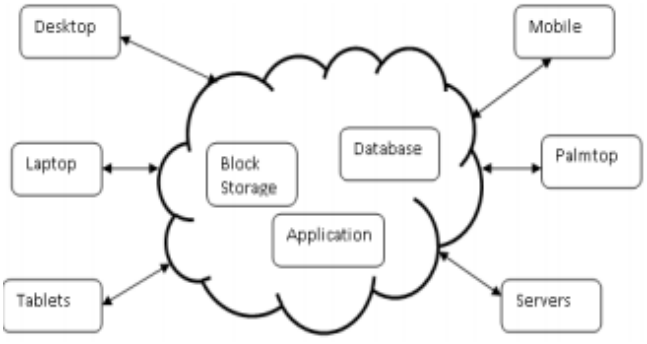


Fig1: Example diagram for data sharing with cloud storage.

To achieve secure data transaction in cloud, suitable cryptography method is used. The data owner must encrypt the file and then store the file to the cloud. If a third person downloads the file, he/she may view the record if he/she had the key which is used to decrypt the encrypted file. Sometimes this may be failure due to the technology development and the hackers. To

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

overcome the problem there are lot of techniques introduced to make secure transaction and secure storage. The encryption standards used for transmit the file securely. The assured deletion technique aims to provide cloud clients an option of reliably destroying their data backups upon requests. The encryption technique was implemented with set of key operations to maintain the secrecy. Recently, Sushmita ruj [1] addressed Anonymous Authentication [1] for data storing to clouds. Anonymous authentication is the process of validating the user without the details or attributes of the user. So the cloud server doesn't know the details or identity of the user, which provides privacy to the users to hide their details from other users of that cloud. Security and privacy protection in clouds are examined and experimented by many researchers. Wang et al. [16] provides storage security using Reed-Solomon erasure correcting codes. Using homomorphism encryption, [17] the cloud receives cipher text and returns the encoded value of the result. The user is able to decode the result, but the cloud does not know what data it has operated on. Time-based file assured deletion, which is first introduced in [5], means that files can be securely deleted and remain permanently inaccessible after a predefined duration.

The main idea is that a file is encrypted with a data key by the owner of the file, and this data key is further encrypted with a control key by a separate key manager (known as Ephemerizer [5]). The key manager is a server that is responsible for cryptographic key management. In [5], the control key is time-based, meaning that it will be completely removed by the key manager when an expiration time is reached, where the expiration time is specified when the file is first declared. Without the control key, the data key and hence the data file remain encrypted and are deemed to be inaccessible. Thus, 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. An open issue in the work [5] is that it is uncertain that whether time-based file assured deletion is feasible in practice, as there is no empirical evaluation. Later, the idea of time-based file assured deletion is prototyped in Vanish [15]. Vanish divides a data key into multiple key shares, which are then stored in different nodes of a public Peer-to-Peer Distributed Hash Table (P2P DHT) system. Nodes remove the key shares that reside in their caches for a fixed time period. If a file needs to remain accessible after the time period, then the file owner needs to update the key shares in node caches. Since Vanish is built on the cache-aging mechanism in the P2P DHT, it is difficult to generalize the idea from time-based deletion to a finegrained control of assured deletion with respect to different file access policies.

We propose policy based file access [2] and policy based file assured deletion [2], [5], [7] for better access to the files and delete the files which are decided no more. We propose effective renewal policy for making better approach to renew the policy without downloading the data key and control keys, which is available now a day. Instead we can add a renew key with each file and download that keys whenever the file needs to be renewed.

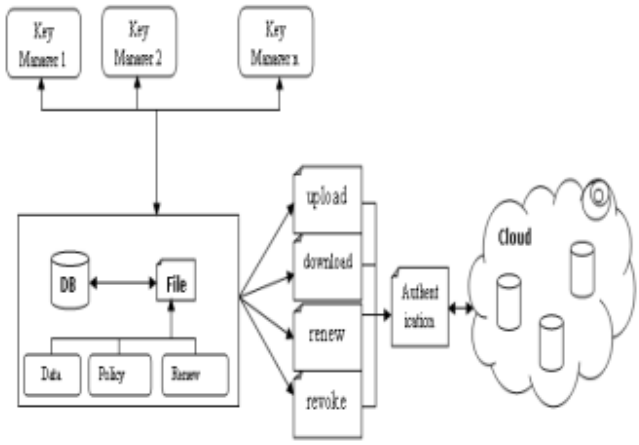


Fig2: Overall system diagram.

First the client was authenticated with the username and password, which is provided by the user. Then the user was asked to answer two security levels with his/her choice. Each security levels consist of 5 user selectable questions. The user may choose any one question from two security levels. The private key for encrypt the file was generated with the combination of username, password and the answers for the security level

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

questions. After generating the private key the client will request to the key manager for the public key. The key manager will verify the policy associated with the file. If the policy matches with the file name then same public key will be generated. Otherwise new public key will be generated. With the public key and private key the file will be encrypted and uploaded into the cloud.

If a user wants to download the file he/she would be authenticated. If the authentication succeeded, the file will be downloaded to the user. Still the user cant able to read the file contents. He / she should request the public key to the key manager. According to the authentication, the key manager will produce the public key to the user. Then the user may decrypt the file using the login credentials given by the user and the public key provided by the key manager. The client can revoke the policy and renew the policy due to the necessity.

2. KEY MANAGEMENT In this paper, following are the cryptographic keys to protect data files stored on the cloud Public Key: The Public key is a random generated binary key, generated and maintained by the Key manager itself. Particularly used for encryption/ decryption. Private Key: It is the combination of the username, password and two security question of user's choice. The private key is maintained by client itself. Used for encrypt / decrypt the file. Access key: It is associated with a policy. Private access key is maintained by the client. The access key is built on attribute based encryption. File access is of read or write. Renew key: Maintained by the client itself. Each has its own renew key. The renew key is used to renew the policy of each necessary file at easy method.

3. PROPOSED WORK

Encryption / Decryption We used RSA algorithm for encryption/Decryption. This algorithm is the proven mechanism for secure transaction. Here we are using the RSA algorithm with key size of 2048 bits. The keys are split up and stored in four different places. If a user wants to access the file he/she may need to provide the four set of data to produce the single private key to manage encryption/decryption. B. File Upload / Download 1. File Upload

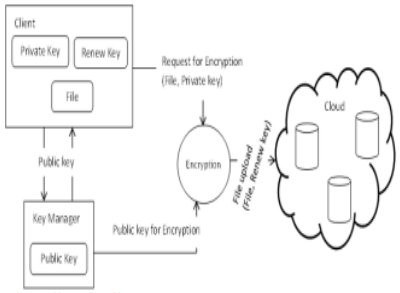


Fig3: File uploading process.

The client made request to the key manager for the public key, which will be generated according to the policy associated with the file. Different policies for files, public key also differs. But for same public key for same policy will be generated. Then the client generates a private key by combining the username, password and security credentials. Then the file is encrypted with the public key and private key and forwarded to the cloud.

2. File Download The client can download the file after completion of the authentication process. As the public key maintained by the key manager, the client request the key manager for public key. The authenticated client can get the public key. Then the client can decrypt the file with the public key and the private key. The users credentials were stored in the client itself. During download the file the cloud will authenticate the user whether the user is valid to download the file. But the cloud doesn't have any attributes or the details of

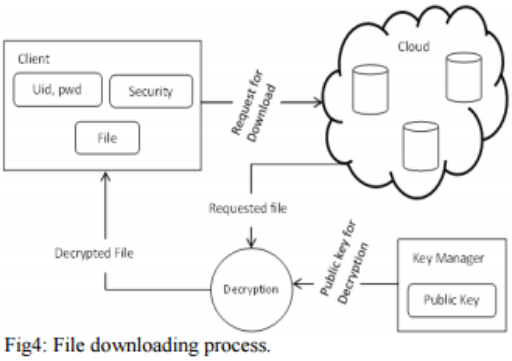


Fig4: File downloading process.

the user.

C. Policy Revocation for File Assured Deletion The policy of a file may be revoked [8] under the request by the client, when expiring the time period of the

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

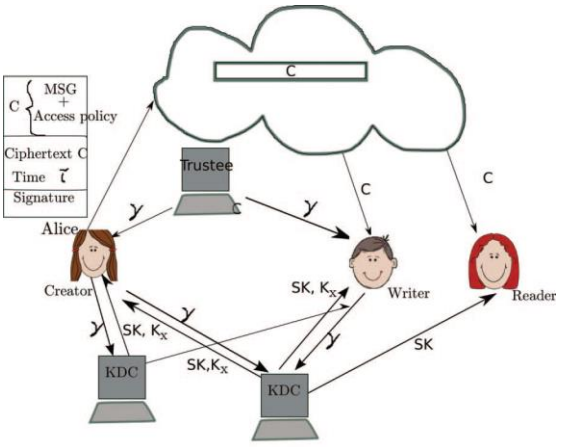
contract or completely move the files from one cloud to another cloud environment. When any of the above criteria exists the policy will be revoked and the key manager will completely removes the public key of the associated file. So no one recover the control key of a revoked file in future. For this reason we can say the file is assuredly deleted. Automatic file revocation [12] scheme is also introduced to revoke the file from the cloud when the file reaches the expiry and the client didn't renew the files duration. D. File Access Control Ability to limit and control the access to host systems and applications via communication links. To achieve, access must be identified or authenticated. After achieved the authentication process the users must associate with correct policies with the files. To recover the file, the client must request the key manager to generate the public key. For that the client must be authenticated. The attribute based encryption standard is used for file access which is authenticated via an attribute associated with the file. With file access control the file downloaded from the cloud will be in the format of read only or write supported. Each user has associated with policies for each file. So the right user will access the right file. For making file access the attribute based encryption scheme is utilized.

E. Policy Renewal Policy renewal is a tedious process to handle the renewal of the policy of a file stored on the cloud. Here we implement one additional key called as renew key, which is used to renew the policy of the file stored on the cloud. The renew key is stored in the client itself.

4.PROPOSED PRIVACY PRESERVING AUTHENTICATED ACCESS CONTROL SCHEME

In this section, we propose our privacy preserving authenticated access control scheme. According to our scheme a user can create a file and store it securely in the cloud. This scheme consists of use of the two protocols ABE and ABS, as discussed in Sections 3.4 and 3.5, respectively. We will first discuss our scheme in details and then provide a concrete example to demonstrate how it works. We refer to the Fig. 1. There are three users, a creator, a reader, and writer. Creator Alice receives a token _ from the trustee, who is assumed to be honest. A trustee can be someone like the federal government who manages social insurance numbers etc. On presenting her id (like health/social insurance number), the trustee gives her a token _. There are multiple KDCs (here 2), which can be scattered.

For example, these can be servers in different parts of the world. A creator on presenting the token to one or more KDCs receives keys for encryption/decryption and signing. In the Fig. 1, SKs are secret keys given for decryption, Kx are keys for signing. The message MSG is encrypted under the access policy X. The access policy decides who can access the data stored in the cloud. The creator decides on a claim



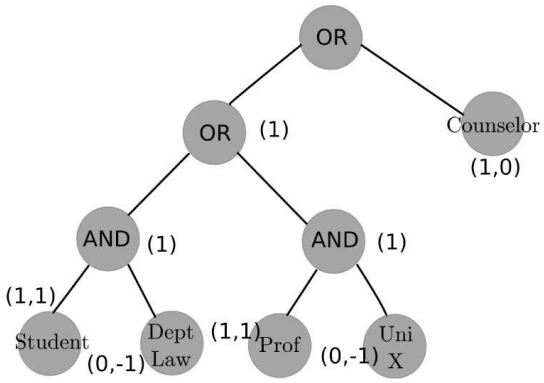
policy Y, to prove her authenticity and signs the message under this claim. The ciphertext C with signature is c, and is sent to the cloud. The cloud verifies the signature and stores Our secure cloud storage model the ciphertext C. When a reader wants to read, the cloud sends C. If the user has attributes matching with access policy, it can decrypt and get back original message Write proceeds in the same way as file creation. By designating the verification process to the cloud, it relieves the individual users from time consuming verifications. When a reader wants to read some data stored in the cloud, it tries to decrypt it using the secret keys it receives from the KDCs. If it

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

has enough attributes matching with the access policy, then it decrypts the information stored in the

5 REAL LIFE EXAMPLE

We now revisit the problem we stated in the introduction. We will use a relaxed setting. Suppose Alice is a law student and wants to send a series of reports about



malpractices by authorities of University X to all the professors of University X, Research chairs of universities X; Y ;Z and students belonging to Law department in university X. She wants to remain anonymous, while publishing all evidence. All information is stored in the cloud. It is important that users should not be able to know her identity, but must trust that the information is from a valid source. For this reason she also sends a claim message which states that she "Is a law student" or "Is a student counselor" or "Professor at university X." The tree corresponding to the claim policy is shown in Fig. 2. The leaves of the tree consists of attributes and the intermediary nodes consists of Boolean operators. In this example the attributes are "Student," "Prof," "Dept Law," "Uni X," "Counselor." The above claim policy can be written as a Boolean function of attributes as

6 SECURITY OF THE PROTOCOL

In this section, we will prove the security of the protocol. We will show that our scheme authenticates a user who wants to write to the cloud. A user can only write provided the cloud is able to validate its access claim. An invalid user cannot receive attributes from a KDC, if it does not have the credentials from the trustee. If a user's credentials are revoked, then it cannot replace data with previous stale data, thus preventing replay attacks.

7 COMPUTATION COMPLEXITY

In this section, we present the computation complexity of the privacy preserving access control protocol. We will calculate the computations required by users (creator, reader, writer) and that by the cloud. Table 2 presents notations used for different operations.

The creator needs to encrypt the message and sign it. Creator needs to calculate one pairing eðg; gÞ. Encryption takes two exponentiations to calculate each of $C_{1;x}$. So this requires $2mE_T$ time, where m is the number of attributes. User needs to calculate three exponentiation to calculate $C_{2;x}$ and $C_{3;x}$. So time taken for encryption is ð3m þ 1ÞE$_0$ þ $2mE_T$ þ $_P$. To sign the message, Y ; W; S_i^0s and P_js have to be calculated as well as HðCÞ. So, time taken to sign is ð2l þ 2ÞE$_1$ þ 2tE$_2$ þ $_H$. The cloud needs to verify the signature. This requires checking for (11). Time taken to verify is ðl þ 2t$Þ_P^\wedge$ þ lðE$_1$ þ E$_2$Þ þ $_H$. To read, a user needs only to decrypt the

8 COMPARISON WITH OTHER ACCESS CONTROL SCHEMES IN CLOUD

We compare our scheme with other access control schemes and show that our scheme supports many features that the other schemes did not support. 1-W-M-R means that only one user can write while many users can read. M-W-M-R means that many users can write and read. We see that most schemes do not support many writes which is supported by our scheme. Our scheme is robust and decentralized, most of the others are centralized. Our scheme also supports privacy preserving authentication, which is not supported by others. Most of the schemes do not support user revocation, which our scheme does. In Tables 4 and 5, we compare the computation and communication costs incurred by the users and clouds and show that our distributed approach has comparable costs to centralized approaches. The most expensive operations involving pairings and is done by the cloud. If we compare the computation load of user during

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

read we see that our scheme has comparable costs. Our scheme also compares well with the other authenticated scheme.

9 CONCLUSION

We have presented a decentralized access control technique with anonymous authentication, which provides user revocation and prevents replay attacks. The cloud does not know the identity of the user who stores information, but only verifies the user's credentials. Key distribution is done in a decentralized way. One limitation is that the cloud knows the access policy for each record stored in the cloud. In future, we would like to hide the attributes and access policy of a user.

10ACKNOWLEDGEMENTS

I am **NAMPALLY SUSHMA** and would like to thank the publishers, researchers for making their resources material available. I am greatly thankful to Associate Prof: **MRS.B.MANASA** for their guidance. We also thank the college authorities, PG coordinator and Principal for providing the required infrastructure and support. Finally, we would like to extend a heartfelt gratitude to friends and family members

11.REFERENCES

[1] S. Ruj, M. Stojmenovic, and A. Nayak, "Privacy Preserving Access Control with Authentication for Securing Data in Clouds," Proc.

IEEE/ACM Int'l Symp. Cluster, Cloud and Grid Computing, pp. 556- 563, 2012.

[2] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward Secure and Dependable Storage Services in Cloud Computing," IEEE Trans. Services Computing, vol. 5, no. 2, pp. 220-232, Apr.- June 2012.

[3] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy Keyword Search Over Encrypted Data in Cloud Computing,"Proc. IEEE INFOCOM, pp. 441-445, 2010.

[4] S. Kamara and K. Lauter, "Cryptographic Cloud Storage," Proc. 14th Int'l Conf. Financial Cryptography and Data Security, pp. 136- 149, 2010.

[5] H. Li, Y. Dai, L. Tian, and H. Yang, "Identity-Based Authentication for Cloud Computing," Proc. First Int'l Conf. Cloud Computing (CloudCom), pp. 157-166, 2009.

[6] C. Gentry, "A Fully Homomorphic Encryption Scheme," PhD dissertation, Stanford Univ., http://www.crypto.stanford.edu/craig, 2009.

[7] A.-R. Sadeghi, T. Schneider, and M. Winandy, "Token-Based Cloud Computing," Proc. Third Int'l Conf. Trust and Trustworthy Computing (TRUST), pp. 417-429, 2010.

[8] R.K.L. Ko, P. Jagadpramana, M. Mowbray, S. Pearson, M. Kirchberg, Q. Liang, and B.S. Lee, "Trustcloud: A Framework for Accountability and Trust in Cloud Computing," HP Technical Report HPL-2011-38,

Miss NAMPALLY SUSHMA. MTech student, in M.Tech Student, of CSE in **Amr Institute of Technology**, mavala, T.S, India

Mrs.B.MANASA working as a Associate Professor at **Amr Institute of Technology**, mavala, T.S, India, Graduate from JNTUH Hyderabad. She has 2 years of UG/PG Teaching Experience