ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 11, November 2013

Secure Cloud Storage using Decentralized Access Control with Anonymous Authentication

R.Ranjith[1], D.Kayathri Devi[2]

PG Student, ANNA UNIVERSITY, Nodal Center- Kamaraj College Of Engineering & Technology,

Virudhu nagar, Tamilnadu, India[1]

Assistant Professor, Depatrment of IT, Kamaraj College Of Engineering & Technology,

Virudhu nagar, Tamilnadu, India[2]

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**.**

I. 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 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.

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

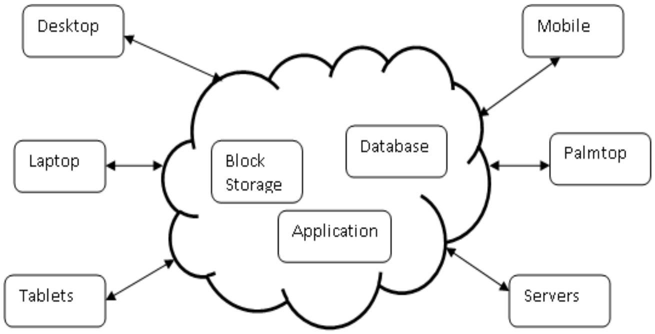

Fig1: Example diagram for data sharing with cloud storage.

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 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 homomorphic 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 fine-grained 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.

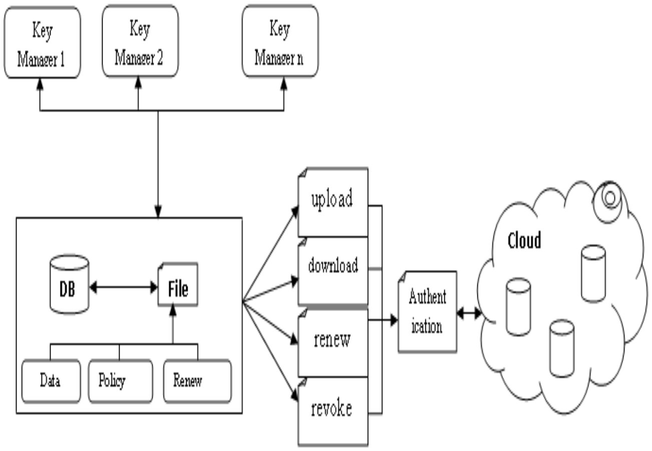

Fig2: Overall system diagram.

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 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.

II. 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.

III. PROPOSED WORK

A. *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*

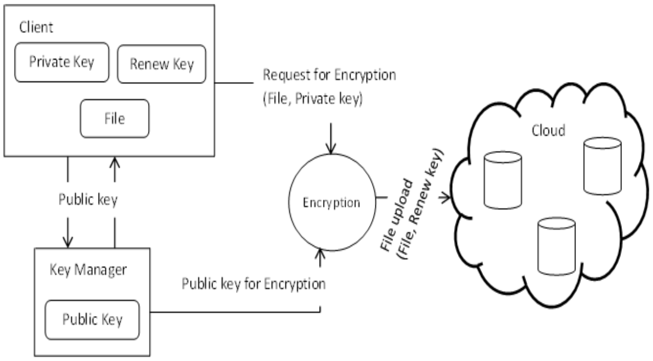

Fig3: File uploading process.

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 the user.

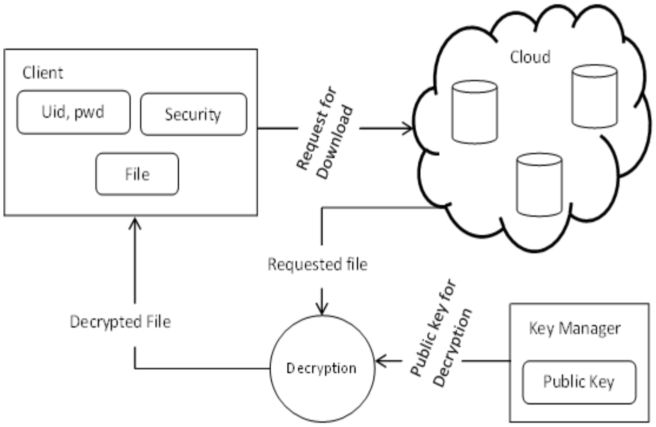

Fig4: File downloading process.

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 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.

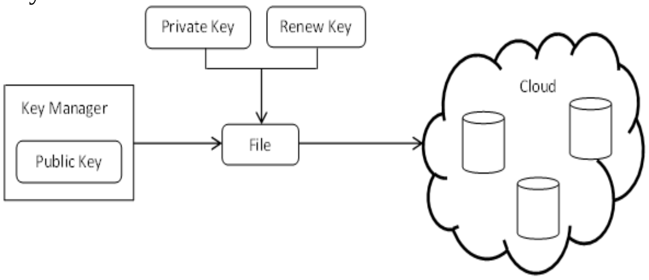

Fig5: file uploading with data key and renew key.

When a file's contract time reaches to expire or a policy has to be revoke on the cloud, there is no need to download all the keys from the cloud. Instead of one renew key is used to revoke the policy. The client creates a renew key for each file and the keys are encrypted with the control key and fetched with the files, then sent to the cloud.
The renewal can be done by the following steps:

1. Download all the encrypted renew keys of each file from the cloud.
2. Send the renew keys to the key manager for decrypt the renew key with the control key.
3. Get the renew keys from the key manager.
4. Generate new renew keys and encrypts with control key.
5. Send the renew keys to the cloud to make the policy renewal of each file.

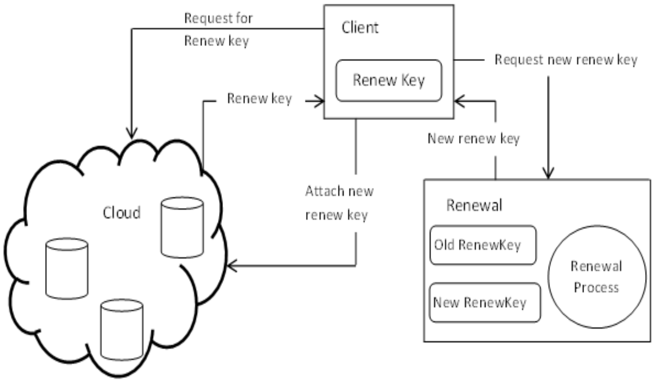

Fig6: Policy Renewal process.

IV. Performance Analysis

A. Time Performance

The performance of this paper was analysed under various file sizes. At first the time performance of this paper is evolved for different file sizes. Then the cryptographic operation time is evolved. The only achievement of this paper is, it supports random time duration for any size of files to download.

Table1: Time Performance for transaction on cloud

FileSize	Upload (sec)	Download (sec)
10bytes	15	0
1kb	17	3
10kb	19	0
100kb	20	7
1mb	22	7

B. Upload

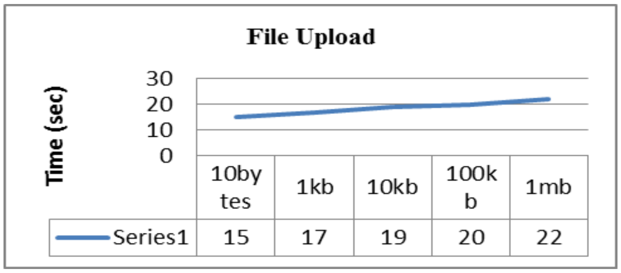

Fig7: Performance Analysis of File Upload Process

File uploading time is not a constant one. For same size file the time taking for uploading is randomly different. Using the time taken to upload the file one can identify the encryption standard. To confuse the hacker the random time delay is achieved.

C. Download

File downloading time is also not a constant one. For same size file the time taking for downloading is randomly different. Using the time taken to download the file one can identify the encryption standard. To confuse the hacker the random time delay is achieved.

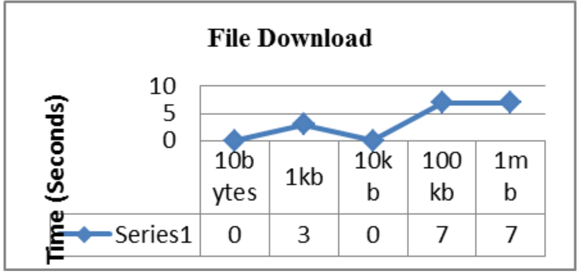

Fig8: Performance Analysis of File Download Process

V. Conclusion

We propose secure cloud storage using decentralized access control with anonymous authentication. The files are associated with file access policies, that used to access the files placed on the cloud. Uploading and downloading of a file to a cloud with standard Encryption/Decryption is more secure. Revocation is the important scheme that should remove the files of revoked policies. So no one can access the revoked file in future. The policy renewal is made as easy as possible. The renew key is added to the file. Whenever the user wants to renew the files he/she may directly download all renew keys and made changes to that keys, then upload the new renew keys to the files stored in the cloud.

In future the file access policy can be implemented with Multi Authority based Attribute based Encryption. Using the technique we can avoid the number of wrong hits during authentication. Create a random delay for authentication, so the hacker can confuse to identify the algorithm.

REFERENCES

[1] S Sushmita Ruj, Milos Stojmenovic and Amiya Nayak, "Decentralized Access Control with Anonymous Authentication of Data Stored in Clouds", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS

[2] Yang Tang, Patrick P.C. Lee, John C.S. Lui and Radia Perlman, "Secure Overlay Cloud Storage with Access Control and Assured Deletion", IEEE Transcations on dependable and secure computing, VOL. 9, NO. 6, NOVEMBER/DECEMBER 2012

[3] G. Wang, Q. Liu, and J. Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," in ACM CCS, , pp. 735–737, 2010

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

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

[6] Ruj, A. Nayak, and I. Stojmenovic, "DACC: Distributed access control in clouds," in IEEE TrustCom, 2011

[7] A. Rahumed, H.C.H. Chen, Y. Tang, P.P.C. Lee, and J.C.S. Lui, "A Secure Cloud Backup System with Assured Deletion and Version Control," Proc. Third Int'l Workshop Security in Cloud Computing, 2011

[8] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute Based Data Sharing with Attribute Revocation," Proc. Fifth ACM Symp. Information, Computer and Comm. Security (ASIACCS), Apr. 2010

[9] W. Wang, Z. Li, R. Owens, and B. Bhargava, "Secure and Efficient Access to Outsourced Data," Proc. ACM Workshop Cloud Computing Security (CCSW), Nov. 2009

[10] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," Proc. IEEE Symp. Security and Privacy, May 2006

[11] R. Geambasu, J.P. John, S.D. Gribble, T. Kohno, and H.M. Levy, "Keypad: Auditing File System for Mobile Devices," Proc. Sixth Conf. Computer Systems (EuroSys), Apr. 2011

[12] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Storage Security in Cloud Computing," Proc. IEEE INFOCOM, Mar. 2010

[13] Amazon, "Case Studies," http://aws.amazon.com/solutions/case-studies/ #backup, 2012

[14] Amazon S3, http://aws.amazon.com/s3, 2010

[15] R.Geambasu, T.Kohno, A.Levy and H.M.Levy, "Vanish: Increasing Data Privacy with Self-Destructing Data", Proc. 18th Conf.USENIX Security Symp, Aug. 2009

[16] Wang, Q.Wang, K.Ren, N.Cao and W.Lou, "Toward Secure and Dependable Storage Services in Cloud Computing", IEEE T.Services Computing, Vol. 5, no.2, pp. 220-232, 2012

[17] C.Gentry, "A fully homomorphic encryption scheme", Ph.D. dissertation, Stanford University, 2009, http://www.crypto.stanford.edu/craig

BIOGRAPHIES

R.Ranjith- Received the Master Degree in Computer Applications from Indira Gandhi National Open University of india in 2006 and the M.Phil. degree in Vinayaka Missions University of India in 2008. Currently he is doing Master of Engineering in Computer Science at Anna University of India. His research interests include cloud computing and cloud security.

D.Kayathri Devi- Received the Bachelor of Technology in Information Technology from Anna University of india in 2005 and the Master of Engineering in Computer Science from Anna University of India in 2009. Currently she is doing Ph.D in Information and Communication Engineering at Sethu Institute of Technology of India. She is working as an Assistant Professor in Kamaraj College of Engineering and Technology, Tamilnadu, India. Her research interests include Network Security and cloud security.