ISSN:2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

Secured Resource Sharing in Cloud Storage using Policy based Access Control

S.Imavathy[1], R.Uma Maheswari[2]
[1]Assistant Professor,Dept.Of CSE, E.G.S.Pillay Engineering College,Nagapattinam,Tamilnadu,India
[2]PG Scholar,Dept.Of CSE, E.G.S.Pillay Engineering College,Nagapattinam,Tamilnadu,India

Abstract— Cloud computing is a general term anything that involves delivering hosted services, Anything as a Service (AaaS), over the web on demand basis. It uses the web and central remote servers to maintain data and applications. The lack of confidence in trusting information flow(users data are usually processes remotely in unknown machines that do not owned or operated by user) in cloud has become common, as users fears of losing control of their own data (like personal, professional, financial, Health). In this approach, a secured cloud storage system that achieves policy-based access control is proposed with an information accountability cloud framework to keep track of the actual usage of the clients data.The access policy generated for the file controls the file accesses and policy revocation makes the file permanently inaccessible. The system is built up on a set of cryptographic key operations that are self- maintained by a set of key managers and adds security features. The access details of the data are logged and auditing also performed.

Keywords— cloud computing, policy, logging, auditing, data sharing.

I. INTRODUCTION

Cloud computing is an emerging technology and it is purely based on internet and its envirnonment and presents a new way to supplement the current consumption and delivery model for IT services based on the Internet, by providing for dynamically scalable and often virtualized resources as a service over the Internet. This technology allows for much more efficient computing by centralizing storage, memory, processing and bandwidth. Cloud computing is broken down into three segments: "application" "storage" and "connectivity". Each segment serves a different purpose and offers different products for businesses and individuals around the world. . It allows for easy connectivity to servers and information sharing. It assures for appropriate use of resources as the users are required to pay only for the services they require and it is highly reliable & redundant.

A cloud system may have many cloud service providers (CSPs) to improve the performance. Based on availability and work load, the system selects a CSP for the client accessing it. Hundreds or thousands of clients may access the system simultaneously; hence the availability is a major problem. It can be improved by CSPs with data replication. Since the data and data owner is same for all copies of the data, this causes another concern of accountability. So the owner has to aware about the copies of data and who all are using it. Since the data is storing remotely; the data owner is unaware of where the data is located and how many copies are created. This may lead to unauthorized access of the data. The data owner may want to set some restrictions to clients who are trying to access the data. In this scenario, the distributed data should keep all these details. But again the authorized clients should be categorized according to permission; it will be a problem in a distributed system with many clients.

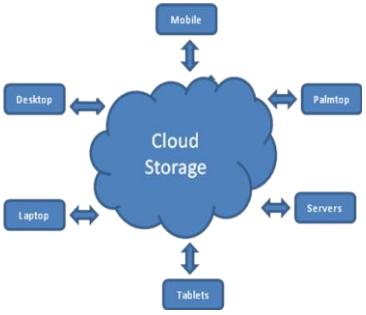

Fig1: Example diagram for Data Sharing with Cloud

A cloud framework is created for secured data sharing, can be called as cloud information accountability.This performs automated logging and distributed auditing of relevant access performed by any entity, carried out at any point of time at any cloud service provider. Cloud Framework has two major components: logger and log harmonizer. The logger is the component which is strongly coupled with the owner's data, so that it records each and every access to the data by any other client. The log harmonizer forms the central component which allows the data owner to access the log files. The security is again a major concern. To overcome these problems, a new approach is proposed for secured data sharing with authentication.

The system allows clients to be timely and accurately informed about their data usage. The proposed system allows image sharing in a secured manner. The data owner can share images with access permissions. Each image/file has some access control and only authorized clients can access the images/files. Assured deletion of files, which promise permanent deletion from the storage, is another feature. To provide guarantees of access control and assured deletion, cryptographic schemes including threshold secret sharing and attribute based encryption (ABE) are using. This paper presents secured cloud storage for image sharing with access policies. The clients of the system must satisfy certain access policies to access the images/files stored. The files are stored along with file access policies and for assured deletion the corresponding policies are revoked and hence the files are permanently inaccessible for clients. The access information is recorded and the owners can check it.

In this paper, section II discusses the related works and some techniques used for security. The section III explains system overview and different components of the system architecture. Section IV deals with Key Management and section V says about the implementation of the cloud framework with automated logging and auditing. Section VI discusses the performance evaluation and section VII concludes the work so far.

II. RELATED WORK

In this section we review some related works concerned with security and privacy issues in cloud. Also, we briefly discuss the work which adopt similar techniques as our approach but serve for different purposes.

1) Security and Privacy issues in cloud

We put more and more of ourselves in the cloud every day. E-mail, device settings, data synchronization between devices, and access to, much of our digital selves is tied to a handful of cloud service accounts with Google, Gmail, Dropbox, and others etc. Security and Privacy Risks for the related data in Cloud Computing also increases. Concern arise as in cloud it is not always clear to an individual why their personal information is requested or how it will be used or passed on to others parties. Till today, little work has been done regarding accountability and auditing in cloud and lot to be researched.

Pearson et al. have proposed accountability mechanisms to address privacy concerns of end users [3] and then develop a privacy manager [2]. Their basic idea is that the user's private data are sent to the cloud in an encrypted form, and the processing is done on the encrypted data. The output of the processing is processed by the privacy manager to reveal the correct result. However, the privacy manager provides only limited features in that it does not guarantee protection once the data are being disclosed. In [6], the authors present a layered architecture for addressing the end- to-end trust management and accountability problem in federated systems. The authors' focus is very different from ours, in that they mainly leverage trust relationships for accountability, along with authentication and anomaly detection. Further, their solution requires third-party services to complete the monitoring and focuses on lower level monitoring of system resources.

A detailed analysis of the cloud computing security issues and challenges is discussed in [6], which is focusing on the cloud computing types and the service delivery types and helps to understand the behavior of clients/users. That mainly proposes the core concept of secured cloud computing which suggests the cloud computing based on separate encryption and decryption services. Even though it suggests methods to avoid threats, solutions for the same are not specified. The proposed architecture uses some important security services including authentication, encryption and decryption.

2)Compression

The same is discussed along with compression in [10]. Key Policy Attribute-Based Encryption (KP- ABE), Proxy Re-Encryption (PRE) and Lazy re-encryption [10] handles many of the security issues. A main issue in the proposed system is distributed auditing. A flexible distributed storage integrity auditing mechanism, utilizing the 'homomorphic' token and distributed erasure-coded data is referred in [11]. The design allows users to audit the cloud storage with very lightweight communication and computation cost. The auditing result not only ensures strong cloud storage correctness guarantee, but also simultaneously achieves fast data error localization. Considering the cloud data are dynamic in nature, the design [11] further supports secure and efficient dynamic operations on outsourced data, including block modification and deletion Users' fears of losing control of their own data can become a significant barrier to the wide adoption of cloud services. This problem is addressed in [5] and they propose a novel highly decentralized information accountability framework to keep track of the actual usage of the users' data in the cloud. Identity Based Encryption [12] Techniques are used for authentication and data security.

In particular, an object-centred approach that enables enclosing our logging mechanism together with users' data and policies also proposed. The common attacks are copying attack, disassembling attack, man-in-the-middle attack and compromised JVM attack [5]. The main issues are overhead added by JVM integrity checking, authentication delay and storage overhead due to large log files. Another technique suggested for secured storage and authentication so far is Attribute Based Encryption [13]. The client based authentication requires access to key and policy of every client, which limits the scalability and flexibility. Attribute based encryption is the appropriate solution for scalability issue.

III. PROPOSED WORK

A Cloud Information Accountability Framework is used for automated logging and distributed auditing of accesses performed by any entity.

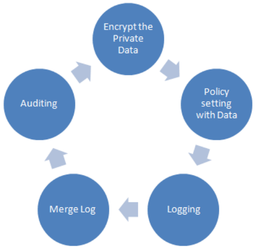

Fig 2: Phases of Cloud Information Accountability Framework.

Having an awareness of the key accountability phases will not only simplify the problem, but also allow tool makers and their customers to gauge the comprehensiveness of tools (i.e. whether there are any phases not covered by a tool). A classification of the different phases may also help researchers to focus on specific research sub-problems of the large cloud accountability problem. These phases are collectively known as the Cloud Information Accountability Framework., which consists of the following five phases (see Fig: 2)

1) Encrypt the Private data: The user's private data are sent to the cloud in an encrypted form, and the processing is done on the encrypted data.

2) Policy Setting with Data: The policies are stored in the cloud storage along with corresponding identity attribute. The client can access the file only if the access policy satisfies with file access policy.

3)Logging: Considerations include the lifespan of the logs within the cloud, the detail of data to be logged and the location of storage of the logs.

4)Merge Logs: The owner receives log files periodically and he/she can find out unauthorized accesses easily.

5)Auditing: Logs and reports are checked and potential irregularities highlighted. The checking can be performed by auditors or stakeholders.

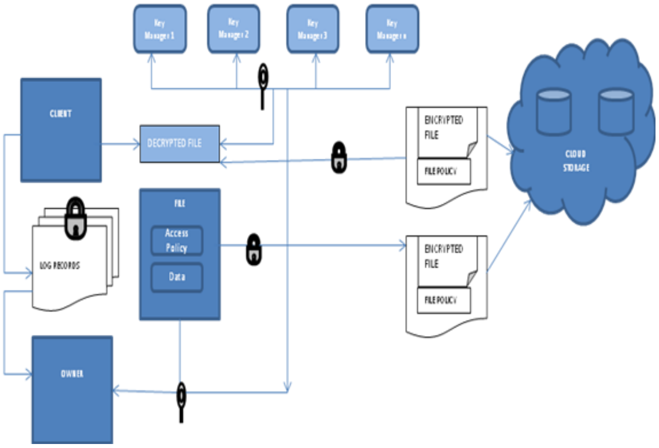

Fig 3:System Overview

The clients are requested to register before sharing and accessing their images. The data owners upload their images with certain access policies. The images are encrypted for secured storage and kept in the cloud along with necessary keys. The keys are shared among a set of key managers. On the other hand, clients other than data owners request for images. The clients must satisfy certain access policies to access an image. The log records generated automatically during the access, which is also kept in encrypted format.

A. Access Policy Generation

Policies are created for access control. The clients have to register before accessing the system. An access control policy, for example, client-A can read and download files, is created for each client along with an access key. Similarly, each file also has a file access policy and unique control key. For example, file B can be viewed by clients from India only. The policies are stored in the cloud storage along with corresponding identity attribute. The client can access the file only if the access policy satisfies with file access policy.

B. File Upload

Each file has a file policy and associated control key. For each policy i, the key manager generates two secret large RSA prime numbers p_i and q_i and computes the product $n_i = p_iq_i$. The key manager then randomly chooses the RSA public-private control key pair (e_i, d_i). The parameters (n_i, e_i) will be publicized, while d_i is securely stored in the key manager. For uploading, the client first requests the public control key (n_i, e_i) of policy Pi from the key manager. Then the client generates two random keys K and S_i, and sends {K}S_i, S_i^{ei} and {F}K to the cloud. Then the client must discard K and S_i.

C. *File Download*

The design is based on blinded RSA, in which the client requests the key manager to decrypt a blinded version of the encrypted data key. If the associated policy is satisfied, then the key manager will decrypt and return the blinded version of the original data key. The client can then recover the data key. The motivation of using this blinded decryption approach is, the actual content of the data key remains confidential to the key manager as well as to any attacker that sniffs the communication between the client and the key manager easily. The client fetches {K}S_i, S_i^{ei}, and {F}K from the cloud. Then the client generates a secret random number R, computes R^{ei}, and sends S_i^{ei}. R^{ei} = $(S_iR)^{ei}$ to the key manager to request for decryption. The key manager then computes and returns $((S_iR)^{ei})^{di}$= S_iR to the client, which can now remove R and obtain S_i, and decrypt {K}S_i and hence {F}K.

D. File Deletion

A file will be deleted (or permanently inaccessible) if its associated policies are revoked and become obsolete. That is, even if a file copy that is associated with revoked policies exists, it remains encrypted and we cannot retrieve the corresponding cryptographic keys to recover the file. Thus the file copy becomes unrecoverable by anyone (including the owner of the file). If a policy Pi is revoked, then the key manager completely removes the private control key di and the secret prime numbers P_i and q_i . Thus, we cannot recover S_i from S_ie_i, and hence cannot recover K and file F. We say that file F, which is tied to policy P_i, is assuredly deleted. Note that the policy revocation operations do not involve interactions with the cloud.

E. *Log Records*

Log record is created for each access to images/files. The log record contains the id of the client/client accessed, image/file id, access policy of the client/client, access type, date, time and owner of the image. The logging mechanism is automated and distributed. The client can access images from any of the service providers. Each time the log record is stored in the log file. When a client trying to access an image, it is provided only after checking the permissions granted. Log file is encrypted using a random key and stored in the cloud. The key is again encrypted using access key of the owner. The access key is generated using the access permissions and identity attributes of the owner. The owner receives log files periodically and he/she can find out unauthorized accesses easily.

IV. KEY MANAGEMENT

The system proposes a policy based file/image sharing and policy revocation for file assured deletion. The key managers are responsible for cryptographic key management. The main feature is that a file is encrypted using a data key by the owner of the file, and this data key is further encrypted by a control key by a separate key manager. Without the control key, the data key and hence the data file remain encrypted and are deemed to be inaccessible.

A. *Cryptographic Keys*

1) Data Key: The data key is a random secret key used to encrypt/decrypt files/log records using symmetric key encryption.

2) Control Key: The control key is associated with a particular file policy. It is a public-private key pair and the private control key is managed by a quorum of key managers. It is used to encrypt/decrypt data keys associated with files.

3) Access Key: The access key is associated with access policy of a client/owner. It is used to encrypt/decrypt data keys associated with log files.

B. *Secret Key Sharing*

In key sharing first create N key shares for a key, such that any $M \leq N$ of the key shares can be used to recover the key. To access files associated with active file policies, at least M out of N key managers required to keep the key shares of the required control keys. Then, to assuredly delete files, at least N - M + 1 out of N key managers must securely erase the corresponding control keys of the revoked policies. The parameters M and N determine the tradeoffs between the fault tolerance assumptions of key managers when accessing and deleting files. If M is small (large), then we need fewer (more) key managers to be active in order to access a file, but we need more (fewer) key managers to purge the revoked control keys in order to delete a file.

V. IMPLEMENTATION

The system is divided into many components to provide efficient storage and computing. The two scenarios, the admin and the client are separated in a gentle way and connected to appropriate functions. Cloud Framework is created based on the notion of information accountability.

This framework performs automated logging and distributed auditing of relevant access performed by any entity, carried out at any point of time at any cloud service provider. It provides service for sharing the images of the admin and client. Simply the owner (client) shares data with certain access permissions provided by the system. The clients' with those permissions granted by the system can access the images. The log information is stored and can be viewed by owner.

The entire system is controlled by admin. Admin act as data owner also. The main functions of admin are checking the service request of the client and approving or rejecting the service. Only after confirmation, clients are permitted to log in to the system. The services are limited to the permission granted. Six kinds of permissions are allowed. They are 1) Read Only2) Read and Download3) Limited Time Read 4) Limited Time Download5) Location Read 6) Location Download. Admin as well as clients share images by uploading new images and setting access permissions to the images which are the same as mentioned above. The images can be deleted by the owner only. Log records are generated for each access. The client id and access policy is tracked for the log generation. Later this can be checked the owner.

Policy generation and policy revocation are two main functions. Each client of the system has an access policy and each file is uploaded with some file access policy. The policy is revoked for making the file inaccessible, which is termed as file assured deletion. A unique access key is generated for each client to protect identity information and log information. For secured image/file sharing, a unique random data key and associated control key is generated. A set of key managers keep the secret keys and provides whenever necessary.

C. Security

This system prevents the following two types of attacks. First, an attacker may try to evade the auditing mechanism by storing the files remotely, corrupting the file, or trying to prevent them from communicating with the owner. Second, the attacker may try to compromise the JRE used to run the files.

VI. PERFORMANCE EVALUATION

Performance of the system should be evaluated with all the constraints and requirements. As the system provides cloud computing service, the system should satisfy the advantages of that. The throughput and security should reach some reasonable level. The main functions are logging and auditing. The processing of service request needs the attention of admin. The requests can be handled using admin for each service provider to improve the performance speed. Some performance constraints are listed below,

1) Service Confirmation Time: The time taken to grant a service request. If many users try to send request at same time or many users are waiting for confirmation, it will be a bottleneck for the CSP.

2)Log Creation Time: Finding out the time taken to create a log file when there are entities continuously accessing the data, causing continuous logging is a constraint.

3) Authentication Time: The overhead can occur is during the authentication of a CSP. If the time taken for this authentication is too long, it may become a bottleneck for accessing the enclosed data.

A. Service Time

When many users send service request, with one service provider and admin, it is difficult to check and confirm all the requests without delay. The user has to wait for a reasonable time for getting account confirmation. It is the same, when many clients try to access the image files simultaneously.

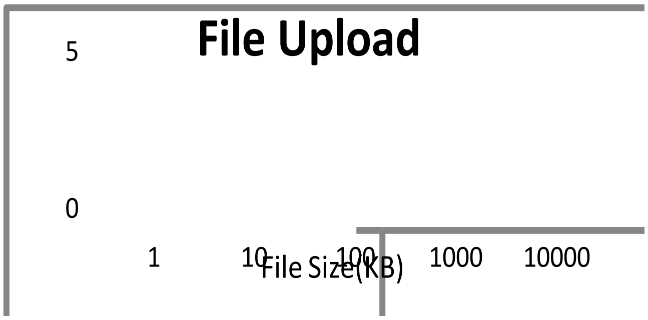

Fig 4: File Uploading [File size Vs Time] graph

Fig 5:File Downloading [File size Vs Time] graph

B. Cryptographic Operation Time

The total time for cryptographic operations includes the total computational time used for performing AES and RSA on the file and data key, and the time for the client to coordinate with the quorum of key managers on operating the cryptographic keys. For a client, the access to the system needs password decryption, data key decryption and file decryption. Similarly, an owner has to go through password decryption, control key generation, data key generation, file encryption and data key encryption. Hence both client and owner take a reasonable time for their operations.

C. Security

This system prevents the following two types of attacks. First, an attacker may try to evade the auditing mechanism by storing the files remotely, corrupting the file, or trying to prevent them from communicating with the owner. Second, the attacker may try to compromise the JRE used to run the files.

VII. CONCLUSION

The system proposed here provides a secured cloud storage system, which provides policy based file access and deletion. The policy will be generated for each client and file. The client, who satisfies the policies of the file, can access the file. A logging mechanism records the access information and auditing mechanism provides this information to the owner. The files and log records are kept as encrypted in the cloud storage system for avoiding various attacks. File assured deletion can be done by policy revocation, which removes the control key of the file from the database and hence the decryption of file is impossible. The functions of various cryptographic keys and operations are discussed along with key sharing. The performance evaluation reveals the constraints. The proposed system will be useful in image sharing when third party cloud storage is used.

References

[1] A.Squicciarini, S.Sundareswaran and D.Lin, “Preventing Information Leakage from Indexing in the Cloud," Proc. IEEE Int'l Conf. Cloud Computing, 2010.

[2] S.Pearson , Y. Shen, and M. Mowbray," A privacy Manager for Cloud Computing," Proc. Int'l Conf. Cloud Computing (cloudcom), pp.90-106,2009

[3] S.Pearson and A. Charlesworth, "Accountability as a Way Forward for Privacy Protection in the Cloud, " Proc First Int'l conf. Cloud Computing, 2009.

[4] R. Corin, S. Etalle, J.I. den Hartog, G. Lenzini, and I. Staicu, “A Logic for Auditing Accountability in Decentralized Systems,” Proc. IFIP TC1 WG1.7 Workshop Formal Aspects in Security and Trust, pp. 187-201, 2005.

[5] Smitha Sundareswaran, Anna C. Squicciarini and Dan Lin, "Ensuring Distributed Accountability for Data Sharing in the Cloud,", IEEE Transaction on dependable a secure computing, VOL. 9, NO. 4, pg 556-568, 2012.

[6] Chun and A. C. Bavier ,"Decentralized Trust Management and Accountability in Federated System," Proc. Ann. Hawaii Int'l Conf. System Science (HICSS), 2004.

[7] B.Crispo and G. Ruffo, “Reasoning about Accountability within Delegation,” Proc. Third Int’l Conf. Information and Comm. Security (ICICS), pp. 251-260, 2001.

[8] S Sundareswaran, A. Squicciarini, D. Lin, and S. Huang, “Promoting Distributed Accountability in the Cloud,” Proc. IEEE Int’l Conf. Cloud Computing, 2011.

[9] Kulkarni G; Dept. of Electron. & Telecommun., Marathwada Mitra Mandal's Polytech., Pune, India, Gambhir J, Patil T and Dongare A, “A security aspects in cloud computing,” Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference

[10] S Sajithabanu and Dr. E George Prakash Raj, Dept. of Computer Science, Bharathidasan University, Trichy, Tamilnadu, India, “Data Storage Security in Cloud ,” IJCST Vol. 2, Issue 4, Oct . - Dec. 2011

[11] Cong Wang, Dept. of Electr. & Comput. Eng., Illinois Inst. of Technol., Chicago, IL, USA , Qian Wang, Kui Ren, Ning Cao and Wenjing Lou, “Toward Secure and Dependable Storage Services in Cloud Computing,” IEEE Transactions on Services Computing, VOL 5, Issue 2, April-June 2012

[12] Cong Wang, Dept. of Electr. & Comput. Eng., Illinois Inst. of Technol., Chicago, IL, USA , Qian Wang, Kui Ren, Ning Cao and Wenjing Lou, “Toward Secure and Dependable Storage Services in Cloud Computing,” IEEE Transactions on Services Computing, VOL 5, Issue 2, April-June 2012

[13] Boneh and M.K. Franklin, “Identity-Based Encryption from the Weil Pairing,” Proc. Int’l Cryptology Conf. Advances in Cryptology, pp. 213-229, 2001

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