SECURITY ENABLED DEDUPLICATION IN CLOUD STORAGE

[1]Raghuram A S, [2]Prakruthi S, [3]Arpitha D, [4]Kiran B
[1]Assistant Professor, [2]Assistant Professor, [3]Assistant Professor, [4]Assistant Professor
[1]Computer Science and Engineering,
[1] ATME College of Engineering, Mysuru, India

Abstract: The cloud storage has become very popular in these days due its advantages like sharing of data among different geographical locations. But in most of the organizations, the storage systems can contain many duplicate copies of many pieces of same data. For example, same file can be saved by the different users using different names of the same document, or two or more files that do not match the name sometimes will be consisting of same as that of previous file. This removes extra by keeping copy, making other to point to original file. Companies regularly uses this process in case of backup and disaster recovery applications, but this is also used to free up space in storage as well. To avoid this duplicate data and to maintain confidentiality in cloud there are many methods proposed. To protect important data while supporting datadeduplication, many more procedures have been proposed.

INTRODUCTION

Cloud computing is way for providing unlimited virtualized resources to the cloud users in the form of service which they are connected across the Internet, with hiding platform and the implementation details.

In recent days clouds service providers (CSP) provides huge capacity of storage and also parallel computing at lower costs .Due to raise in popularity there is a significant increase in storage.Csp provides credentials for customer by providing details.
Data management in varying storage is huge challenge it can be done if information is scalable. It is for removing extra file of similar data information in storage Data compression to enhance utilization and decrease sent bytes through network.
Rather maintaining multiple data copies with same data, this process eliminates information by keeping only one physical file and then referring other copied data to that copy.

LITERATURE PAPERS

"Twin cloud architecture"

Datadeduplication is for eliminate extra copy of same making link to new user, in this he is considered as owner to that data. In this approach owner who uploads data first generates hash value based on document using function FileTag (File) - It computes sha-1 hash of File as FileTag.

Proof of data by owner

In this paper they achieve reduplication by checking proof of owner. At time of transmitting a file to cloud it is provided with set of privileges so that privileged user can access file or check duplicate files. Files downloaded with user details before verifying same document. User gets copy if present and has privilege access.

Message locked Encryption:

Convergent inscription method afford confidentiality of data to user's outsourced information that will be maintained on public clouds. In this scenario encapsulating key got from message. It supports datadeduplication, because when we use duplicate file, generate ditto key so it will get equal cipher text which makes datadeduplication possible.

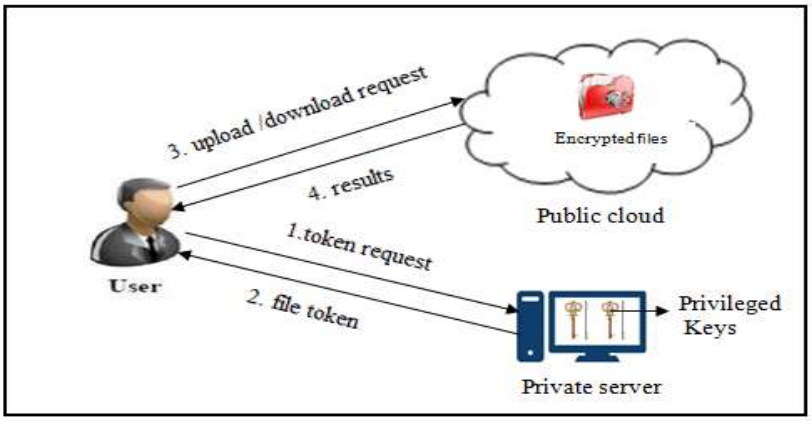
System Design

The proposed system firstly generates a hash value by using merkle tree algorithm, then it consists of two types of encryption first is the symmetric encryption for encrypting file Second is asymmetric encryption which is used for encrypting key.

MerkleTree Algorithm

In proposed project, hash value created ,from this is used. There are many steps in merkle tree where in each SHA-256 is applied to generate hash values This SHA-256 comes under SHA-2. SHA-256 is function in cryptography generates values Usually SHA-256 create particular and fixed size hashes in this values cannot be decoded.

Merkletree is used for generating hash for a document. In this system it is used for generating hash values for all type of files which is used for comparing with existing hash values in the cloud.



Architecture of Proposed System

Rijndael Algorithm

This practice uses same key for encapsulation of information. This application uses this for encryption.Rijndael comes under advanced schemas. Earlier practices do not provide effective schemes.

The problem with previous methods is that if consecutive terms are same then key generated for that letter is also same. By this cyberpunk easily predict information, to overcome this problem Rijndael is used, uses secret key that declared in code. Once secret key is found it is added with salt data.combination of both constitutes key. Initializationvector is defined for encrypting initial block or letter. That initialization vector is generated by creating an instance to it.

IMPLEMENTATION

After designing necessary algorithms, usecases, sequence diagram system has to be developed. To implement this application, requires 3 hosts in network, First system is private server that is handled by admin. This system is used to register user userid, password Admin creates two keys and stores in xml. To store the data of a member xml files are used because that doesn't require extra software like sql. That makes software light weight,Queries are used to take information from xml, once registered member he can access cloud.

Second system is member application, where member login information is present in private server. Data present in another system taken through class termed remoting.Memberid, password is inserted to get logged in,by clicking login button following steps takes place. First carriers are created then it is registered, Once channel is registered interaction is made by specifying type of connection, IP address of private server where data is present, and last field is port number, With this channel it interacts with private server. After establishing connection it checks for particular member information if it is present then navigates to home form else it prints a message login details not found.

Home form consists of text box for inserting a file to transfer, browse button is used for selecting a file from existing local directories, reset button is used clearing text box. Below this there is list view box used to output registered users .The user who logged is showed by checking check box initially. For producing hash value member has to browse the file in to textbox by clicking browse button.

Hash built for browsed document .It opens and partition in to equal sizes,for every chunk value of hash is created. Next to concatenate consecutive chunks, and then generate hash code for it,this is repeated till single value these values stored in a tag attribute, Tag is a property where they store data but not visible. After generating hash now task it to place in cloud. Before transferring member has to decide for which user file must be shared,this is done using checking a checkbox in list view.

Initially only a member logged in check box will be checked. Now member tick checkbox to provide information, by this selected files are shared. Member will click transfer button, by clicking it these steps occur. To share document firstly it ticks checkbox, if none is checked message box is displayed with message select one member id for sharing file, after picking users to share file upload is clicked.

When upload button is pressed it checks proxy cloud with the hash value. Proxy cloud is a system which is used to store all information about member id of user who stored a file, Hash value and size of file. After checking with proxy cloud it returns file is present or not. If file is present then it generates convergent key using hash value and links member id with file that is present in proxy cloud. If not, it generates convergentkey from hash value, and then it uploads a file in to cloud. After placing in cloud it returns message in message box document transferred successfully. Proxy is updated with storage name, where it is value, size, member id and date time, now member will become owner. This holder will request public key from all checked users, that data is stored in private server. Once this key is returned by private server, encrypt convergent using with this public key and stored in private server.

In member page there is another segment called view cloud files. This section stores all the file uploaded by user and also consists of shared files. If a member wishes to take file, just click on button. When download button is clicked it takes encrypted file from cloud, then it takes encrypted convergent key that is stored in private server and decrypts it. After decrypting a file message box is displayed with message, file downloaded successfully stored in a path, Member can delete if he is a holder of that. If this is shared then if he tries to delete it displays a message if file is uploaded from another member then only that link is deleted but file is deleted from cloud.

CONCLUSION

A simple yet effective method is used for datadeduplication and to minimize storage. Where hash code of each information is stored and compared with new one, if it is already present then discard it and make link to user otherwise add to storage. Furthermore encrypted key is stored in private server which is accessible only to privileged users, using that he/she gets data and use it. This can be applied at level of block in future work.

REFERENCES:

[1] P. Anderson and L. Zhang. Fast and secure laptop backups with encrypted de-duplication. In Proc. of USENIX LISA, 2010.
[2] M. Bellare, S. Keelveedhi, and T. Ristenpart. Dupless: Serveraided encryption for deduplicated storage. In USENIX Security Symposium, 2013
[3] M. Bellare, S. Keelveedhi, and T. Ristenpart. Message-locked encryption and secure deduplication. In EUROCRYPT, pages 296–312, 2013.
[4] M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. J. Cryptology, 22(1):1–61, 2009.
[5] M. Bellare and A. Palacio. Gq and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In CRYPTO, pages 162–177, 2002.
[6] J. Li, X. Chen, M. Li, J. Li, P. Lee, andW. Lou. Secure deduplication with efficient and reliable convergent key management. In IEEE Transactions on Parallel and Distributed Systems, 2013.

.