To Protect Confidentiality, Reliability and Achieve De-duplication In Cloud Storage System.

Khushboo Bakade, Apeksha Bodke, Prachi Pawar, Khushali Tatiya

Student,Information Technology,SKNSITS,Maharashtra,India
Student,Information Technology,SKNSITS,Maharashtra,India
Student,Information Technology,SKNSITS,Maharashtra,India
Student,Information Technology,SKNSITS,Maharashtra,India

ABSTRACT

In the cloud storage system, Data De-duplication is widely used to eliminate the duplicate copies of data. This improves the storage capacity and reduce reliability. With the huge number of users there can be number of copies of the same data on the cloud storage. Thus, Data de-duplication system keeps the record of only one copy for each file and provides reference of the file when required. This process therefore introduces the Proof Of Ownership concept. In the cloud storage system, the challenge of privacy for sensitive data is a major concern. Therefore for a secure storage a deterministic secret sharing scheme is introduced. We show that our proposed authorized duplicate check scheme incurs minimal overhead compared to normal operations.

Keyword : - *De-duplication, Proof of Ownership, distributed storage system, reliability, secret sharing, authorized duplicate check, confidentiality, Cloud Architecture.*

1. INTRODUCTION

There is tremendous growth of digital data, de-duplication system contributes in minimizing network and improve storage by eliminating multiple copies of data. It greatly improves Storage utility by restricting the user to upload the same copy of data. Instead It gives the reference of the owner for the same copy of redundant data. De-duplication is been appreciated by both academia and industry. There are number of de-duplication system that are proposed on various strategies such as file-level or block-level. The file-level de-duplication discovers redundancies to reduce capacity demands and block-level de-duplication discovers and removes redundancy between data blocks. The file is divided into small fixed size of block. The fixed size block simplifies the computations of block boundaries.

The de-duplication system reduces the reliability of the system although it improves the storage space. Data reliability is reduced because there is only one copy of the file. If the file is lost then there could be a loss of very large amount of data and there is unavailability of data. To avoid this the files are divided into locks where if the unauthorized user checks in, the data would not be retrieved for the user and there can be affordable loss instead of losing the whole file. Hence, this overcomes the short coming. The uploaded file is encrypted using AES algorithm. Therefore providing confidentiality for the file. The challenge of data privacy is satisfied using the Encryption mechanisms of the content of the file. This mechanism encrypts the data before outsourcing the data to the cloud. Hence this mechanism generates a hash key for the content and name of the file. The generated hash key is unique as per the file content for any particular data. The hash key is further used for file token generation. For duplicate check of the file, file token is used and hash value of stored file is compared. This is the guideline for the de-duplication process.

2. EXISTING SYSTEM

The various kinds of data for each user stored in the cloud and the demand of long term continues assurance of the data safety, the problem of verifying correctness of data storage in the cloud becomes even more challenging. Cloud computing is not just a third party data warehouse. The data stored in the cloud may be frequently updated by the users, including insertion, modification, deletion, appending, re-ordering etc. One critical challenge of today's cloud storage services is the management of the ever increasing volume of data. According to the analysis report of IDC, the volume of data in the wild is expected to reach 40 trillion gigabytes in 2020.The base line approach suffers two critical deployment issues. First is, it is in-efficient as it will generate enormous number of keys with the increasing number of users. Specifically, each user must be associate an encrypted convergent key with each block of its outsourced encrypted data copies so as to later restore the data copies. Although different users may share the same data copies, they must have their own set of convergent keys so that no other users can access their files. Second, the baseline approach is unreliable as it requires each user to dedicatedly to protect his own master key, If the master key is accidently lost then the user data cannot be recovered, if it is compromised by attackers then the user data will be leaked. Introduction related your research work

2.1 Disadvantage of Existing System

Traditional encryption, while providing data confidentiality, is incompatible with data de-duplication. Identical data copies of different users will lead to different ciphertext making de-duplication impossible. In the existing system, the complete file is encrypted providing less security to the file. If the attacker attacks the whole file can be recovered. In proposed system we tend to split the file in n blocks and then encrypt the data.

3. PROPOSED SYSTEM AND OBJECTIVE

In this paper, we show how to design secured de-duplication system with higher reliability in cloud computing. We introduce the distributed cloud storage servers into de-duplication system to provide better fault tolerance. We enhance our system in security. Specifically, we present an advance scheme to support stronger security by encrypting the file with differential privilege key. In this way the users without corresponding privileges cannot perform duplicate check. Further, more such unauthorised users cannot decrypt the ciphertext collude with S_CSP. Security analysis demonstrates that our system is secured in terms of the definition specifies in the proposed security module.to further protect data confidentiality, the secret sharing technique is utilised which is also compatible with the distributed storage systems. In more details the file is first split and encoded into fragments by using the techniques of secret sharing, instead of encryption mechanism. These shares will be distributed across multiple independent storage servers. Further, more to support de-duplication, a short cryptographic hash value of the content will also be computed and sent to each storage server as the finger print of the fragment stored at each server. Only the data owner who first uploads the data is required to compute and distribute such secret share, while all following users who own the same data copy do not need to compute and store these shares anymore. To recover data copies user must access a minimum number of storage servers through authentication and obtain the secret shares to reconstruct the data. In order words, the secret share of data will only be accessible by the authorised user who own the corresponding data copies. Four new secured de-duplication systems are proposed to provide efficient de-duplication with high reliability for file-level and block-level de-duplication respectively. The secret splitting technique, instead of traditional encryption method is utilised to protect data confidentiality. Specifically data splits into fragments by using secure secret sharing scheme and stored at different servers. The advantages of placing decoy in a file system are threefold: i. The detection of masquerade activity. ii. The confusion of attacker and the additional cost incurred to distinguish real from bogus information and , iii. The deterrence effect which, although hard to measure, plays a significant role in preventing masquerade activity by risk- averse attackers.

3.1 Architecture

Data de-duplication is the latest technique as it stores huge amount of data. Data de-duplication identifies the similar object by comparing with another objects and store only dissimilar objects. The system performs de-duplication technique for storing different types of file.

Architecture diagram consists of Data user, private cloud and public cloud. Data user is nothing but the client who wants to upload or download file. User can upload various type of files such as .txt, .jpg, .jpg, .png, etc. but for uploading or downloading the file user need to be authenticate. Hence for authentication user need to register and login. Once the authentication is done, user will be able to upload/download the files. Private cloud are those who customize the services of public cloud. In this system, private cloud middleware for the client and public cloud, private cloud performs functions like generating file token, requesting to public cloud for duplicate check and convergent key generation. Public cloud provides physical hardware for storage and performs complex operations. In our system, we can store the data on public cloud and can perform encryption, decryption, splitting and sharing on cloud

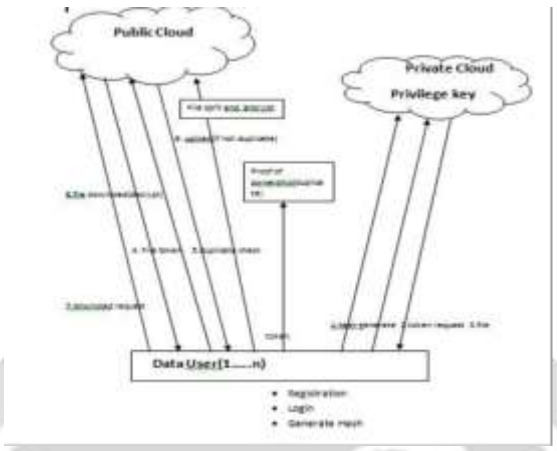


Fig 1:Architecture Diagram

3.2 Algorithms

AES Algorithm steps:
The AES algorithm is used to encrypt and decrypt files which we are going to upload.
1.Derive the set of round keys from the cipher key.
2.Initialize the state array with the block data(plaintext).
3.Add the initial round key to the starting state array.
4.Perform nine rounds of state manipulation.
5.Perform the tenth and the final round of state manipulation.
6.Copy the final state array out as the encrypted data(ciphertext).
SHA 1 Algorithm Steps:
SHA 1 algorithm is used to hash key generation and to check file de-duplication.
1.Derive the set of round keys from the cipher keys.
2.Appending Padding Bits. The original message is ”padded”(extended)so that its length(in bits) is congruent to448,modulo 512.
3. Appending Length.64 bits are appended to the end of the padded message to indicate the length of the original message in bytes.
4. Preparing Processing Functions.
5. Preparing Processing Constants.
6. Initializing Buffers.
7. Processing Message in 512 Blocks.
HMAC SHA Algorithm steps:
The HMAC SHA algorithm is used to token generation which is used for proof of ownership
1. Append zeros to the left end of K to create a b-bit string K+(for example ,if L is length 160bits and b=512,then K will be appended with 44 zero bytes 0x00).
2. XOR(bitwise exclusive OR)k+ with i-pad to produce the b-bit block Si.
3. Append M to Si.
4. Apply H to the stream generated in step 3.
5. XOR K+with opad to produce the b-bit block So.
6. Append the hash result from step 4 to So.
7. Apply H to the stream generated in step 6 and output the result.
Secret Sharing Scheme(SSS):
The secrete sharing algorithm is used for splitting and merging of uploaded file.
i.the sharing algorithm:

Share(M)->(S1,S2,…Sn, pub) the secretes S1,….Sn are distributed securely among servers 1through n, and pub is a public share.(We include pub for the sake of generality but observe that it is often empty. If pub is present, we assume that it is authenticated, so on one can change it: it's just published in the sky.)

ii. The recovery algorithm:

Rec(S'1,S'2….,S'n, pub)=M'. The correctness property of the algorithm m says that for any message M, Rec(Share(M))=M.

3.3 Modules

3.3.1 Secure De-duplication File Checking

Data de-duplication is a specialized data compression technique for eliminating duplicate copies of repeating data. Related and somewhat synonymous terms are unintelligent compression and single-instance storage. This technique is used to improve storage utilization and can also be applied to network data transfers to reduce the number of bytes that must be sent. In the de-duplication process, unique chunks of data, or byte patterns, are identified and stored during a process of analysis. As the analysis continues, other chunks are compared to the stored copy and whenever a match occurs, the redundant chunk is replaced with a small reference that points to the stored chunk. Given that the same byte pattern may occur dozens, hundreds, or even thousands of times, the amount of data that must be stored or transferred can be greatly reduced.

3.3.2 Decoy Documents

We propose a different approach for securing data in the cloud using offensive decoy technology. We monitor data access in the cloud and detect abnormal data access patterns. We launch a disinformation attack by returning large amount of decoy information to the attacker. This protects against the misuse of the user's real data. We use this technology to launch disinformation attacks against malicious insiders, preventing them from distinguishing the real sensitive customer data from fake worthless data. The decoy then serve two purpose:

1.Validating whether data access is authorized when abnormal information access is detected, and

2.Confusing the attacker with bogus information.

3.3.3 Secret Sharing Security

There are two algorithms in a secret sharing scheme, which are Share and Recover. The secret is divided and shared by using Share. With enough shares, the secret can be extracted and recovered with the algorithm of Recover. In our implementation, we will use the Ramp secret sharing scheme(RSSS), to secretly split a secret into shards. Specifically, the(n,k,r)-RSSS (where n>k>r>=0) generates n shares from a secret so that the secret can be recovered from any k or more shares, and no information about the secret can be deduced from any r or less shares. Two algorithms, Share and Recover are defined in the (n,k,r)-RSSS.

3.3.4 User Behavior Profiling

We monitor data access in the cloud and detect abnormal data access patterns User profiling is a wel known technique that can be applied here to model how, when, and how much a user accesses their information in the cloud Such 'normal user' behaviour can be continuously checked to determine whether abnormal access to the user's information is occurring. This method of behaviour-based security is commonly used in the fraud detection applications. Such profiles would naturally include volumetric information, how many documents are typically read and how often. We monitor for abnormal search behaviour that exhibit deviations from the user baseline the correlation of the search behaviour anomaly detection with trap-based decoy files should provide stronger evidence of malfeasance, and therefore improve a detector's accuracy.

3.3.5 Distributed De-duplication System with Tag Consistency

In this section, we consider how to prevent a duplicate faking or maliciously-generated ciphertext replacement attack. A security notion of tag consistency has been formalised for this kind of attack. In a de-duplication storage system with tag consistency, it requires that no adversary is able to obtain the same tag from a pair of different messages with non-negligible probability. This provides security guarantees against the duplicate faking attacks in which a message can be undetectably replaced by a fake one. In the previous related work on reliable de-duplication over the encrypted data, the tag consistency cannot be achieved as the tag is computed by the data owner from underlying data files, which cannot be verified by the storage server.

3.3.6 De-duplication System with Proof of Ownership

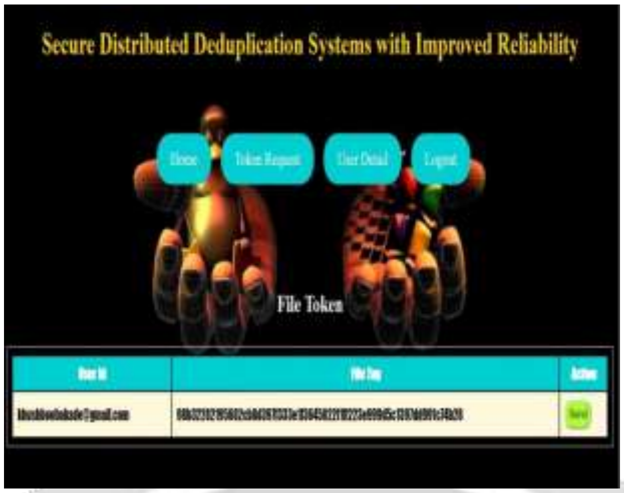
Halevi pointed out the weakness of the security in traditional de-duplication systems with only a short hashing value. Halevi showed a number of attacks that can lead to data leakage in the storage system supporting client-side de-duplication. To overcome this security issue, hey also presented the concept of Proof of Ownership(PoW) to prevent these attacks. PoW enables users to provide their ownership of data copies to the storage server.
To prevent these attacks we proposed the notion of 'proof of ownership' for de-duplication systems, so that a client can efficiently prove to the cloud storage server that one owns a file without uploading the file itself.

3.3.7 Confidentiality

The confidentiality against two types of adversaries. The first type of adversary is defined as dishonest users who aims to retrieve files stored at S-CSPs they do not own. The second type of adversary is defined as a group of S-CSPs and users. Their goal is to get the useful information of the file content they do not own individually by launching the collusion attack. The attacks launched by these two types of adversaries are denoted by Type-1 and Type-2 attack respectively. Because the RSSS is used in our construction, the different level of confidentiality is achieved in terms of the parameter r given in the RSSS scheme, which increases with the number of r.

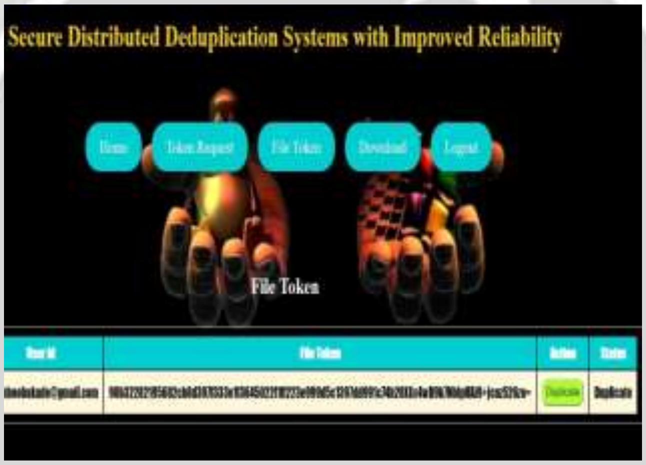
3.3 Figure Caption

Fig.1.Hash code generation


Secure Distributed Deduplication Systems with Improved Reliability
Home
Token Request
User Detail
Logout
File Token

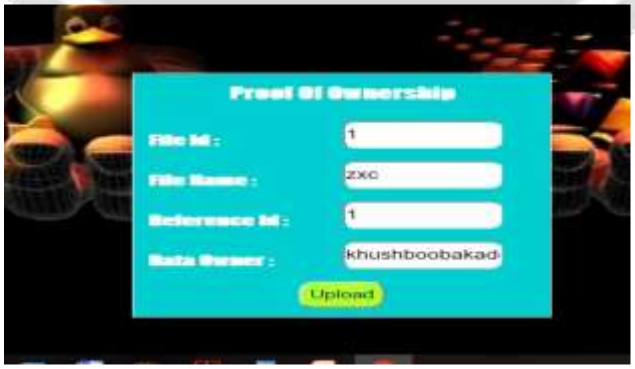
User Id	File Tag	Action
khushboobakade@gmail.com	[illegible]	Send

Fig.2.File token generation


Secure Distributed Deduplication Systems with Improved Reliability
Home
Token Request
File Token
Download
Logout
File Token

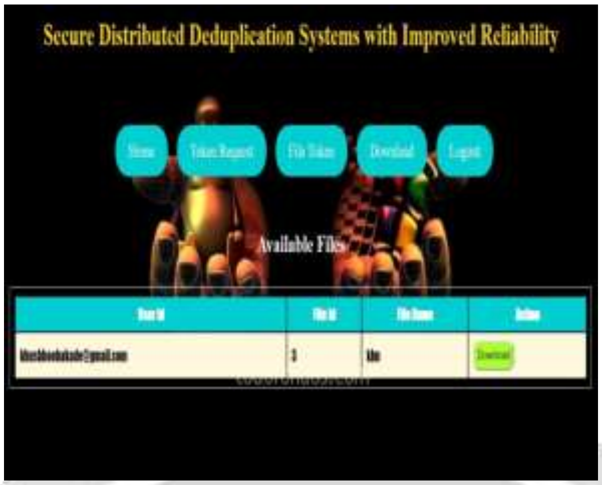
User Id	File Token	Action	Status
boobakade@gmail.com	[illegible]	Duplicate	Duplicate

Fig.3Duplicate check



File Id :	1
File Name :	zxc
Reference Id :	1
Data Owner :	khushboobakad

Fig.4. Upload and PoW



Secure Distributed Deduplication Systems with Improved Reliability

Available Files

User Id	File Id	File Name	Action
[illegible]@gmail.com	3	[illegible]	Download

Fig.5.Downloading original file

4. CONCLUSION

As per proposed system, we have achieved data deduplication with improved reliability and confidentiality with encryption mechanism using secret sharing scheme. We implemented our deduplication systems using Ramp secret sharing scheme. We proposed four construction to support file-level and block-level data deduplication.

6. REFERENCES

[1]. D.Harnik,B.Pinkas,and A.Shulman-Peleg,"Side channels in cloud services:Deduplication in cloud storage."IEEE Security and Privacy,vol.8,no.6,pp.40-47,2010.

[2]. R.D.Pietro and A.Sornotti,"Boosting efficiency and security in proof of ownership for deduplication."in ACM Symposiumon information,Computer and Communications Security,H,Y,Youm and Y.Won,Eid. ACM,2012,pp.81-82

[3]. J.Xu,E-C.Chang,and J.Zhou,"Weak leakage-resilent client-side deduplication of encrypted data in cloud storage,"in ASIACCS,2013,pp.195-206.

[4]. W.K.Ng,Y.Wen and H.Zhu,"Private data deduplication protocols in cloud storage ."in Proceedings of the 27th Annual ACMSymposium on AppliedComputing,S.Ossowski and P.Lecca,Eds.ACM,2012,pp441-446.

[5]. P.Anderson and L.Zhang,"Fast and secure laptop backups with encrypted de-duplicateion,"in Proc.of USENIX LISA,2010.