INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE

ISSN: 2057-5688

Client-End Deduplication on Encrypted Data with Public Auditing in Cloud Storage

DHANDAMUDI HAPPY BALA SOWRI [1], **SD. RESHMA** [2]

[1]Assistant professor, [2] Assistant Professor

CSE Department, Sri *Mittapalli College* of *Engineering*, *Guntur*, *Andhra Pradesh*-522233

Abstract: ***Cloud storage is one of popular cloud service model that stores data on the Internet through the cloud computing provider who manages and provides Storage as a service to the end users. In cloud storage huge amount of data will becomes duplicated due to the multiple uploads from the different users for the same data. The cloud servers want to reduce the volume of data stored by the client which has the similar content and the client wants to maintain the integrity of the data uploaded to the cloud storage. To achieve this there are several models defined using the deduplication and integrity auditing delegation techniques. This paper work is to show an implementation of the combining the deduplication algorithm with the data integrity auditing algorithm to achieve the goals of removing the duplication of data and providing the integrity of data to the client. The data considered will be in encrypted form and the hash data is used to verify the correctness of the data. The proposed algorithm will satisfy the fundamental security requirements and provides an user interface for the client and the server admin to present the proper management of the data stored on cloud storage. The project work is implemented using the java language on Google drive storage.***

Keywords: Cloud storage, deduplication, Data integrity, Google drive, Java, Public auditing algorithm

1. Introduction

Cloud Storage Computing that emerged from a decade is excellent storing platform for many enterprises IT industries. The Cloud vendors are providing a wide form of services as computing, storage, resources and infrastructures through Internet, with scalable features for archiving data in a cost saving manner. Enterprise IT infrastructure can use the Cloud storage to get on-demand storage with no expenditure for hardware. These Cloud service providers are making the Enterprises to have zero maintenance of the archive data. This made the public cloud to have the huge famous for the Enterprise adoption. As per statistics it is observed that over 93% in 2018 from 90% in 2017 of enterprises have started using the public cloud storages. However, there is one issue in cloud storage adoption that is the hidden cost for the transactions that happens during the archiving of the data. The enterprise has to pay the additional cost for the duplication of the data that raises due to the backup process. To avoid this hidden cost the deduplication technique is used.

The use of network storage system is gaining a broader interest due to its cost-effective storage platforms. These platforms present the transmission, storage in multisystem environment and high computing of outsourced data in a pay per use businesses.

For saving the resources consumption in terms of both network bandwidth and storage capacities many of the public service providers such as Dropbox, Google and AWS are applying the client-side deduplication techniques. Data deduplication technique in one way for reducing the cost on cloud storage. For example, an 250GB uncompressed data can be stored in 10GB space if we have and 25% compression of the data. This can even be more depending upon the type of data be backup to the storage. This may save thousands of dollars to the Enterprises that are saving large datasets.

Thus, it is important that the cloud vendor do the deduplicate data so that there will be cost saving for the enterprises. An cloud vendor may save the data in compressed form for example, If a client sends 30 gigabytes of data to the vendor for storage, then the vendor will store that is compressed for 3 gigabytes only. Now the vendor will charge for the 30GB of storage not as 3GB of storage to the client. Thus, the end users will not receive the exposure to the deduplication capabilities that the cloud service provider are using in data storage.

Data deduplication, or “dedupe”, is a data compress technique that removes the duplicate information from a dataset before storing in the server. This method will reduce the space requirement for storing large data sets in the cloud servers.

At a top prospective, the deduplication process will work as the function to remove the repeated data before going to storage. This makes that the server will store only one copy of the data and any other copies will get removed by creating the pointer or reference to the original copy on the place of requirement. This process will work in transparency for end users and cloud service providers. Looking in deep representation of deduplication, the software will generate an unique identifiers for the data using the cryptographic hash function. The file level will be inefficient because the file may get altered during the transmission or during the

storage. A single bit change will make the entire file restored in the cloud.

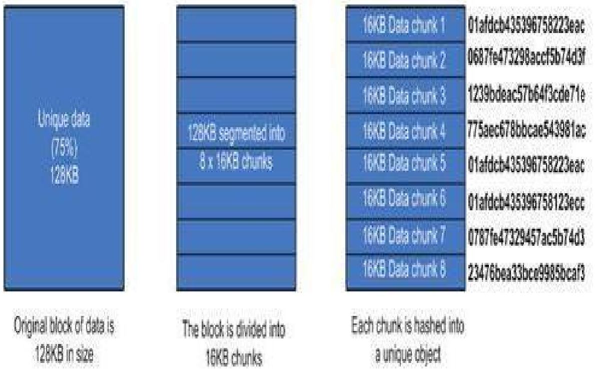


Figure 1: Hash value for unique data block

2. Related Work

Taek-Young Youn et. al,[14] shows a deduplication which implies data sharing, access control permissions in encrypted deduplication storage. This method is effective over traditional storage method. Therefore, the deduplication is combined with data access control method to achieve flexibility of storing data on servers. This work proposes CP-ABE encryption along with data deduplication to solve the problem. It is based on client-side deduplication and provides confidentiality using encryption procedure to private expose of users sensitive data to untrusted cloud servers. The algorithm involves with the convergent encryption scheme with CP-ABE to allow only the authorized users to access critical data. This method provides adequate trade-off between the storage space efficiency and security in cloud environment. Thus, the method is suitable for hybrid cloud models. P Puzio et. al
[13] explain the design of a secure storage service which assures the block-level deduplication and data security. This was designed to consider the scope of increase in number of users and the size of data they are sending on to cloud storage providers. By storing the unique copy of duplicate data, cloud providers will have the great storage reduction and data transfer costs. This work has proposed ClouDedup algorithm which is implemented for block-level deduplication and data confidentiality at the same time. It also shows the encryption operation and an access control mechanism implemented. This has an issue in key management for each block along with the deduplication operation. Its new component will provide an overall impact on the storage and computational costs.

Kim et. al. [12] proposes an new client-side deduplication technique with secure mechanism that prevent a poison attack. The Message-locked encryption (MLE) which is a widespread cryptographic primitive that gives encrypted data stored along with deduplication method on cloud. This work reduces the amount of traffic to be transmitted over the network and requires very few network cryptographic operations to execute. This has promising efficiency over the existing primitive methods based on security, communication costs and computational requirements.

The users of cloud are more concerned with the integrity of data that is stored on the cloud, because the user's data may be vulnerable for a greater number of attacks. The attacks can be by an insider or the outside attackers. Therefore, data auditing is defined, which checks the correctness of the data at any point of time to the user with the help of an trusted process called Third Party Auditor (TPA).

With the invention of Cloud Computing there were many problems came to existence over the privacy and integrity of the user's data that need to be stored on cloud. There is a need of efficient methods to ensure the integrity of data on cloud. Wang et. al. proposed the new method called privacy preserving public auditing protocol that shown the use of Trusted Party Auditing (TPA) to perform the audit of the data on behalf of users. This protocol used the public key based homomorphic linear authenticator (HLA) with random masking technique. But, this method is exposed to the message attacks by malicious cloud attackers. To overcome this a new improved scheme which is more secure was introduced. The public auditing scheme with TPA performs data auditing for the users. It uses random masking for the hiding of data. This method is defined with Boneh- Lynn-Shacham (BLS) signatures along with HLA to proved better security. This is partial implemented on EC2 instance of Amazon web services which has demonstrated higher performance for both public cloud and auditor prospective.

Along the side the Wang et. al. [6] also proposed public auditing schemes and data dynamics by using BLS on HLA with Merkle Hash Tree (MHT). This achieved the integrity of the data but failed in providing the confidentiality of the data that was stored on cloud. [7] To improve the proposed algorithm, they designed an mechanism to detect the modified blocks using homomorphic token based pre-computation to ensure the status change of the users data on the cloud and later to ensure the coded technique to extract the desired block from the cloud.

The Solomon et al. [10] as proposed an method for public auditing scheme for cloud storage in which it is generating a signature set for each file block. The signature set is a collection of signatures in hash form. This as computation and communication overhead. To manage the set for the files stored is not considered in the design and the work not shown how the communication efficiency is achieved.

The Meenakshi et al. [11] are proposed an TPA based protocol to perform the client data auditing using the MHT. This algorithm has the support of data dynamics but it failed to provide the confidentiality of the data stored on the cloud. There is a method which uses the Hash Message Authentication Code (HMAC) keys along with homomorphic tokens to provide the secure data transmission. The verifiability of the data transmitted between the two parties happens using the shared secret key and that key is generated using HMAC based algorithm. Now this work is vulnerable for the attackers to create the fraud messages. Table 2.1. shows the Comparison of Existing Privacy Preserving Public Auditing Scheme. The table shows the comparison of many techniques using many factors and methods. It has the classification where the algorithm supports public auditing, privacy, data dynamics and batch processing. This content also describes where algorithm provides integrity and confidentiality of the data stored on cloud storage. It is clear from the comparison that none of the algorithm provides complete basic data auditing requirements for the assurance of the data security on cloud. Some existing methods have succeeded in providing the privacy preserving and public auditing of the data. But, the algorithm failed to describe the confidentiality of the data that get stored in cloud storage.

Wang et. al[8], The work introduces an securely TPA which is involved in public auditability for the cloud users. The cloud storage is used by the users to store their data in remote location and get benefited of using on-demand high quality applications and services that comes from the shared pool of computing resources without have any drawback of data storage configuration and maintenance. This work will provide a solution for the cloud architecture which has the constrained computing resources. It is used to check the auditability of cloud data with some critical importance so that the users can resort to a third party auditor (TPA). [4] This algorithm is used to check the integrity of outsourced data and establish an error free burden to users. [5] Further, this work performs audits for multiple users simultaneously and efficiently by showing the performance analysis. The drawback of this work is that it is not showing the efficiency achieved during the data transmission medium and it is not including the issues that happen by the attackers during the transmission. Thus requires an improvement to discuss about the data transmission efficiency.

Swapnali More et al. [17] have and improved proposal to establish an secure and efficient privacy based public auditing platform. In this scheme it achieves the privacy of the data stored and provide an efficient mechanism for performing the public auditing on a third party auditor. The data owner uses an mechanism for file splitting into blocks, each block is encrypted using the AES algorithm and for each block it generates SHA-2 hash. The hashes will undergo the signing process using the generated RSA signature. This algorithm involves with the verification process where the signature is generated in TPA and the one stored by the user is used for verifying of the data on the cloud storage. If both the signature get matched it tells that the data is intact and if the signature mismatch then indicates that the integrity failed to the user.

3. Methodology

The system uses the BLS signature-based Homomorphic Linear Authenticator (HLA), which was proposed in [1], for integrity auditing and secure deduplication. The proposed scheme consists of the following entities.

Client (or data owner)
Sends the data to cloud storage. CE-encrypted data is first generated, and then uploaded to the cloud storage to protect confidentiality. The client also needs to verify the integrity of the data.

Cloud Storage Server (CSS).
Provides data storage services to users. Deduplication technology is applied to save storage space and cost. In the project it is assumed that the CSS may act maliciously due

to insider/outsider attacks, software/hardware malfunctions, intentional saving of computational resources, etc.

TPA (Third Party Auditor)

Performs integrity auditing on behalf of the client to reduce the client's processing cost. Instead of the client, the auditor sends a challenge to the storage server to periodically perform an integrity audit protocol. The relation between entities can be seen in figure1.2. A client and a CSS perform PoW for secure deduplication, and a TPA is placed between the client and the CSS to execute integrity auditing instead of the client.

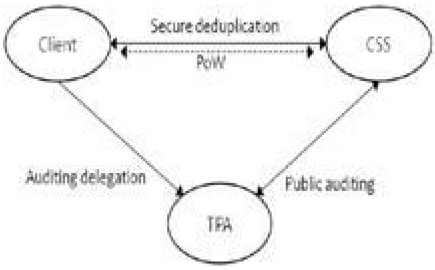


Figure 2: Proposed system architecture

4. Experimental Results

While write buffering and data prefetching techniques reduce interaction with storage devices for data I/O, TPA operations are frequently synchronous.

With the receive for the Client file upload insertion request from the client, the TPA server starts receiving each attributed fields of the file content under creation from the client. The received attributes is stored into the database system. This process flow chart is shown in the figure 3.2.

The conventional file systems will provide a sequential log of operations that allows fast recovery after a failure. In a dedicated TPAS, however, the large number of small updates leads to a pathological I/O profile. The alternative techniques called soft updates require careful ordering of updates, limiting parallelism and placing an higher burden on the storage subsystem.

This project presents a TPA management approach which address the unique performance of a TPA server capable of handling node crashes.

Apps should specify a file extension in the title property during the inserting files with the API. For example, an insert operation for a JPEG file should specify something like "object-id" defined in the file attributes.

ISSN: 2057-5688

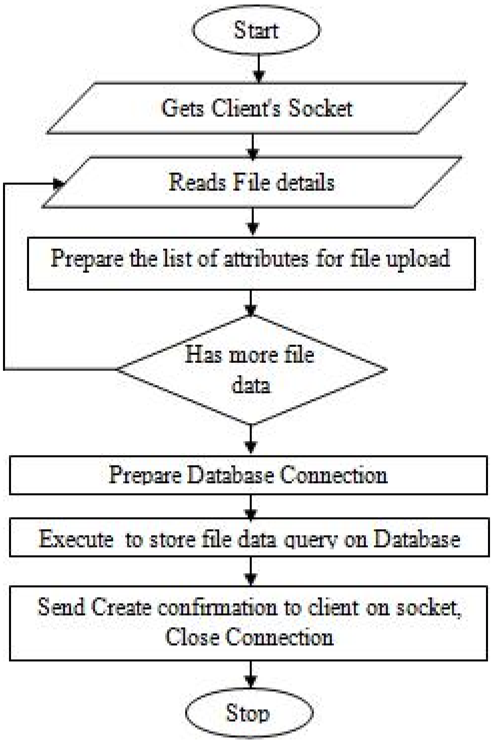

Start
Gets Client's Socket
Reads File details
Prepare the list of attributes for file upload
Has more file data
Prepare Database Connection
Execute to store file data query on Database
Send Create confirmation to client on socket, Close Connection
Stop

Figure 3: Flow Model for the Client file details insertion in TPA server

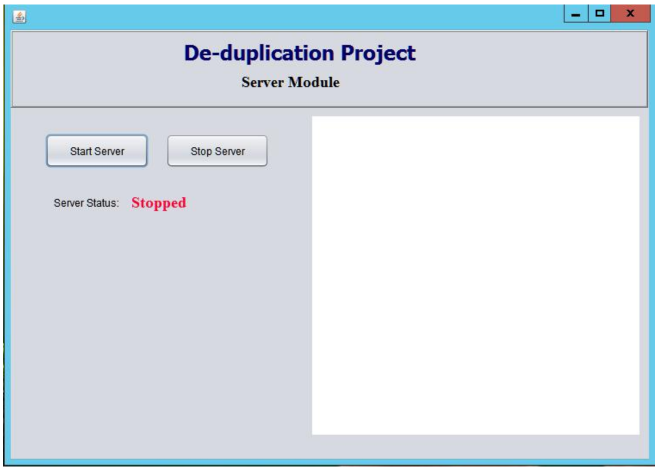
Drive automatically indexes documents for search when it recognizes the file type. This includes text documents, PDFs, images with text, and other common types. If your app saves other types of files (drawings, video, shortcuts), you can improve the discoverability by supplying index able text.

The procedure of authentication and authorization with the client application and Google server will goes with the following steps.

First the software will request the access token from Google server. The access token is received by the Google server will ask for the authentication information. After the successful authentication by the user, for example, the user will connect with the client ID and client secret that is generated in Google development console. Once the authentication is successful the application will receive the session object and an access grant for the Google Drive server. This access session object is used for further sending the queries and commands for Google Drive to perform the needed activity such as file upload, file modify, file delete, folder creation and folder deletion. The file client_secret.json is the content file generated for the client through which the application can read the credentials for the Google drive API. The content of this file looks like

```
{"installed":{
"client_id":"695570159884-csjrpld8vumjobks0dktftisnvp2pkmr.apps.googleusercontent.com",
"project_id":"ninth-iris-241307","
auth_uri":"https://accounts.google.com/o/oauth2/auth",
"token_uri":"https://oauth2.googleapis.com/token",
"auth_provider_x509_cert_url":"https://www.googleapis.com/oauth2/v1/certs",
"client_secret":"gPk7KTUze2QLEG2iyHYaZm7o",
"redirect_uris":["urn:ietf:wg:oauth:2.0:oob","http://localhost"]
}}
```

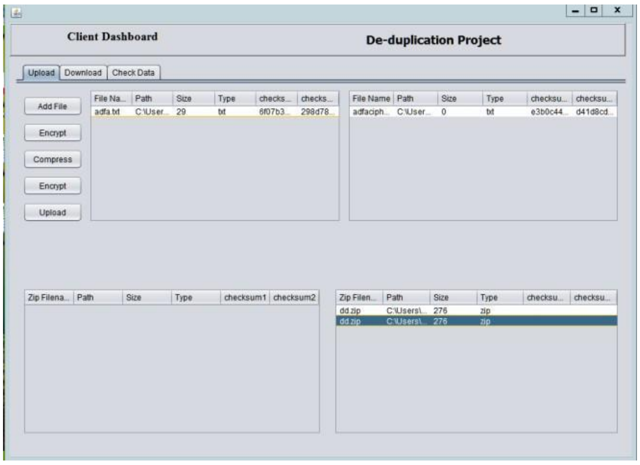
The deduplication project has the dedicated cloud storage server which manages the storing of the user's files into the cloud storage. The figure 5.1 show the graphical user interface created for the server module. The will listen for the user request for the file storage. To start the server start server button is clicked.



De-duplication Project

Server Module

Server Status: Stopped



Client Dashboard

De-duplication Project

File Na...	Path	Size	Type	checks...	checks...
adfa.txt	C:\User...	29	txt	6f07b3...	298d78...

File Name	Path	Size	Type	checksu...	checksu...
adfaciph...	C:\User...	0	txt	e3b0c44...	d41d8cd...

Zip Filena...	Path	Size	Type	checksum1	checksum2

Zip Filen...	Path	Size	Type	checksu...	checksu...
dd.zip	C:\Users\...	276	zip		
dd.zip	C:\Users\...	276	zip		

Figure 5: Cloud storage server status module

The user will starts with login credentials. The user module will have the options to upload the file to the cloud, download the file from the cloud and verify the integrity of the file present in the cloud. figure 5.2 shows the clients dashboard where the interaction of the user with the system is implemented. Each user will go with the corresponding options for performing the need procedure execution.

Figure 4: Simple illustration of single file upload.

5. Conclusion

Emerging Cloud computing and 5G technologies are giving the more data storage requirements for the cloud storage service providers. The cloud storages are used more than ever in the past decade. Managing the storage in an cost effective manner is the primary issue for the cloud storage service providers. This work extends the existing Cloud framework by adding features to cloud storage providers and users. The Deduplication architecture discussed in this work provides a single file system with the traditional sharing and improves on the resource consolidation and scalable performance.

Through this work, an implementation of simple deduplication storage architecture is demonstrated through which shows that the unstructured data files can be encapsulated with an object along with the meta information. The verification of the stored data on cloud can be accomplished using the TPA module.

References

[1] Ibrahim Abaker Tarigo Hashem "The Rise of Big data on cloud Computing: Review and open research issues", 2014 Elsevier.

[2] Prentice Hall "Unstructured Textual data in the organization", Research Paper.

[3] Gollmann. D, Computer Security, 2nd Edition, John Wiley and Sons, 2005.

[4] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Proc. of ESORICS'09, volume 5789 of LNCS. Springer- Verlag, Sep. 2009, pp. 355–370.

[5] Cong Wang, Sherman SM Chow, Qian Wang, Kui Ren, and Wenjing Lou. Privacy Preserving Public Auditing for Secure Cloud Storage. *http://eprint.iacr.org/2009/579.pdf.*

[6] Cong Wang, Sherman SM Chow, Qian Wang, Kui Ren, and Wenjing Lou. Privacy Preserving Public Auditing for Secure Cloud Storage. *Computers, IEEE Transactions on,* 62(2):362–375, 2013.

[7] Cong Wang, Qian Wang, Kui Ren, Ning Cao, and Wenjing Lou. Toward secure and dependable storage services in cloud computing. *Services Computing, IEEE Transactions on*, 5(2):220–232, 2012.

[8] Cong Wang, Qian Wang, Kui Ren, and Wenjing Lou. Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing. *In INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.