RESEARCH ARTICLE OPEN ACCESS

Data DEDUPLICATION Security with Dynamic Ownership Management

MS.C.Kamatchi [1], R.Pooja [2], S.Serishma [3], R.Vanitha [4]
Assistant Professor-II [1], Student [2], [3] & [4]
Department Of Information Technology
Prathyusha Engineering College, Chennai
Tamil Nadu - India.

ABSTRACT
To provide security of deduplication system with higher reliability in which the data are distributed across multiple cloud server.Deduplication technique eliminating redundant data and storing only a single copy of data.By using hash table algorithm, the security requirement of data confidentiality and reducing space storage are also achieved.It converts a range of key values into a range of indexes of an array.It performs a basic operations of a search, insert and deletion.As a result, security analysis demonstrat that our deduplication system are secure in the proposed system. To protect the confidentiality of sensitive data while supporting de duplication, the hash table algorithm technique has been proposed to encrypt the data before outsourcing. To better protect data security, Different from traditional deduplication systems, the differential privileges of users are further considered induplicate check besides the data itself. We also present several new deduplication constructions supporting authorized duplicate check in cloud architecture.

Keywords:- PoW, KP

I. INTRODUCTION

Cloud computing provides seemingly unlimited "virtualized" resources to users as services across the whole Internet, while hiding platform and implementation details. Today's cloud service providers offer both highly available storage and massively parallel computing resources at relatively low costs. As cloud computing becomes prevalent, an increasing amount of data is being stored in the cloud and shared by users with specified *privileges*, which define the access rights of the stored data. One critical challenge of cloud storage services is the management of the ever-increasing volume of data. To make data management scalable in cloud computing, de duplication has been a well-known technique and has attracted more and more attention recently.

Data deduplication is a specialized data compression technique for eliminating duplicate copies of repeating data in storage. The 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. Instead of keeping multiple data copies with the same content, deduplication eliminates redundant data by keeping only one physical copy and referring other redundant data to that copy. Deduplication can take place at either the file level or the block level. For file level deduplication, it eliminates duplicate copies of the same file. Deduplication can also take place at the block level, which eliminates duplicate blocks of data that occur in non-identical files. Although data deduplication brings a lot of benefits, security and privacy concerns arise as users' sensitive data are susceptible to both insider and outsider attacks. Traditional encryption, while providing data confidentiality, is incompatible with data deduplication.

Specifically, traditional encryption requires different users to encrypt their data with their own keys. Thus, identical data copies of different users will lead to different cipher texts, making deduplication impossible. Hash table algorithm has been proposed to enforce data confidentiality while making deduplication feasible.

It encrypts/decrypts a data copy with a convergent key, which is obtained by computing the cryptographic hash value of the content of the data

copy. After key generation and data encryption, users retain the keys and send the cipher text to the cloud. Since the encryption operation is Deterministic and is derived from the data content, identical data copies will generate the same convergent key and hence the same cipher text. To prevent unauthorized access, a secure proof of ownership protocol is also needed to provide the proof that the user indeed owns the same file when a duplicate is found. After the proof, subsequent users with the same file will be provided a pointer from the server without needing to upload the same file. A user can download the encrypted file with the pointer from the server, which can only be decrypted by the corresponding data owners with their convergent keys.

Thus, hash table algorithm encryption allows the cloud to perform deduplication on the cipher texts and the proof of ownership prevents the unauthorized user to access the file. However previous deduplication systems cannot support differential authorization duplicate check, which is important in many applications. In such an authorized deduplication system, each user is issued a set of privileges during system initialization. Each file uploaded to the cloud is also bounded by a set of privileges to specify which kind of users is allowed to perform the duplicate check and access the files. Before submitting his duplicate check request for some file, the user needs to take this file and his own privileges as inputs. The user is able to find a duplicate for this file if and only if there is a copy of this file and a matched privilege stored in cloud. For example, in a company, many different privileges will be assigned to employees.

II. CONTRIBUTION

We propose a deduplication scheme over encrypted data. The proposed scheme ensures that only authorized access to the shared data is possible, which is considered to be the most important challenge for efficient and secure cloud storage services in the environment where ownership changes dynamically. It is achieved by exploiting a group key management mechanism in each ownership group. As compared to the previous deduplication schemes over encrypted data, the proposed scheme has the following advantages in terms of security and efficiency. First, dynamic ownership management guarantees the backward and forward secrecy of deduplicated data upon any ownership change. As opposed to the previous schemes, the data encryption key is updated and selectively distributed to valid owners upon any ownership change of the data through a stateless group key distribution mechanism .The ownership and key management for each user can be conducted by the semi-trusted cloud server deployed in the system. Thus, the proposed scheme delegates the most laborious tasks of ownership management to the cloud server without leaking any confidential information to it, rather than to the users. Second, the proposed scheme ensures security in the setting of by introducing a re-encryption mechanism that uses an additional group key for dynamic ownership group. Thus, although the encryption key (that is the hash value of the file) is revealed in the setting of PoW, the privacy of the outsourced data is still preserved against outside adversaries, while deduplication over encrypted data is still enabled and data integrity against poison attacks is guaranteed.

III. RELATED WORK

Deduplication techniques can be categorized into two different approaches: deduplication over unencrypted data and deduplication over encrypted data. In the former approach, since the hash of the file, which is treated as a “proof” for the entire file, is vulnerable to being leaked to outside adversaries because of its relatively small size. Whereas, in the latter approach, data privacy is the primary security requirement to protect against not only outside adversaries but also inside the cloud server. Thus, most of the schemes have been proposed to provide data encryption, while still benefiting from a deduplication technique, by enabling data owners to share the encryption keys in the presence of the inside and outside adversaries. Since encrypted data are given to a user, data access control can be additionally implemented by selective key distribution after the PoW process. However, not much work has yet been done to address dynamic ownership management and its related security problem.

IV. HASHING TECHNIQUES

Hashing provides very fast to access to records on certain search conditions.this organization is usually called a hash file.the search condition on a single field,called the hash field of the file.the hash field is also called a hash key.the idea behind hashing is to provide a function 'h' called a hash function or randomizing function. That is applied to the hash field value of a recored and yields the address of the disk block in which the record is stored.

TYPES:
1.STATIC HASHING
2.STATIC EXTERNAL HASHING.
3.DYNAMIC AND EXTENDIBLE HASHING TECHNIQUES.

1.STATIC HASHING

A bucket is a unit of storage containing one or more records. The file blocks are divid into m equal sized buckets,numbered Bucket0,bucket1……bucket.in a hash file organizations we obtain the bucket of a record directly from its serarch key value using a hash function.hash function is used to locate records for access,insertion as well as deletion.primary pages fixed .allocated sequentially never de-allocated overflow pages if needed.

H(k) mod M bucket to which data entry with key k belongs (M = # OF BUCKETS)

2.STATIC EXTERNEL HASHING

One of the file fields is designated to be the hash key .k of the file.colisions occur when a new record hashes to a bucket that is already full. an overflow file is kept for storing such records. overflow records that hash to each bucket can be linked to gether to reduce. overflow records uniformly among the buckets .otherwise ,search time will be increased because many overflow records will exits.

3.DYNAMIC AND EXTENDIBLE HASHING TECHNIQUES:

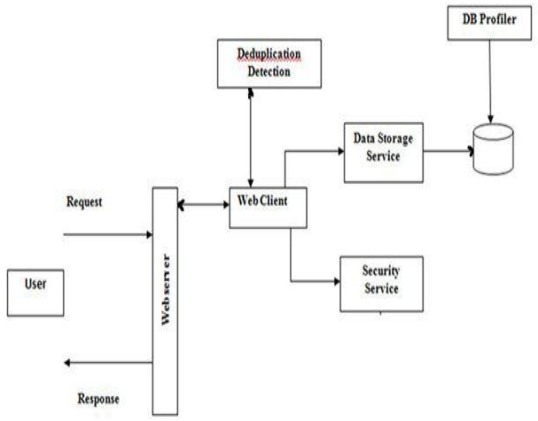
These hashing techniques use the binary representation of the hash value.in dynamic hashing the directory is binary tree. in extensible hashing the diecrtory is an array size 2d where d is called the global depth. The directions can be stored on the disk, and they expand or shrink dynamically .dirctory entries point to the disk blocks . dynamic extendible hashing do not require an overflow area.

V. ENCRPTED AND UNCRYPTED DATA

cloud storage that uses deduplication across multiple users. Specifically, when an attacker temporarily compromises a server and obtains the hash values for data in the cloud storage, he is able to download all these data. This is because only a small piece of information about the data namely its hash value, serves as not only an index of the data to locate information of the data among a huge number of files, but also a "proof" that anyone who knows the hash value owns the corresponding data. Therefore, any users who can obtain the short hash value for specific data are able to access all the data stored in the cloud storage. proposed a randomized threshold to avoid an attack on cloud storage services that which is one of the biggest cloud storage providers. Specifically, the authors showed that spoofing the hash value of a file chunk added to the local Dropbox folder allows a malicious user to access files of other Dropbox users, given that the SHA-256 hash values of the files chunks are known to the attacker. Then, a challenge-response protocol between the server and the client verifies the ownership. PoW is closely related to proof of retrievability and proof of data possession However, proof of retrievability and data possession often use a pre-processing step that cannot be used in the data deduplication procedure. Despite their significant benefits in terms of saving resources, these deduplication schemes may cause another security vulnerability and reveal users' private data, in particular, when partial information of users' data has already been leaked.

VI. SECURE DATA DEDUPLICATION ARCHITECTURE

In this section, we describe the data deduplication architecture and define the security model. According to the granularity



VII. SYSTEM DESCRYPTION AND ASSUMPTION

1) **Data owner:** This is a client who owns data, and wishes to upload it into the cloud storage to save costs. A data owner encrypts the data and outsources it to the cloud storage with its index information, that is, a tag. If a data owner uploads data that do not already exist in the cloud storage, he is called an initial uploader; if the data already exist, called a subsequent uploader since this implies that other owners may have uploaded the same data previously, he is called a subsequent uploader. Hereafter, we refer to a set of data owners who share the same data in the cloud storage as an ownership group.

2) **Cloud service provider:** This is an entity that provides cloud storage services. It consists of a cloud server and cloud storage. The cloud server deduplicates the outsourced data from users if necessary and stores the deduplicated data in the cloud storage. The cloud server maintains ownership lists for stored data, which are composed of a tag for the stored data and the identities of its owners. The cloud server controls access to the stored data based on the ownership lists and manages (e.g., issues, revokes, and updates) group keys for each ownership group as a group key authority. The cloud server is assumed to be honest-but-curious. That is, it will honestly execute the assigned tasks in the system; however, it would like to learn as much information about the encrypted contents as possible. Thus, it should be deterred from accessing the plaintext of the encrypted data even if it is honest.

IMPLEMENTATION

VIII. MODULES

User Module
Server start up and Upload file
Secure DE duplicate System
Download file

Module Description User Module

In this module, Users are having authentication and security to access the detail which is presented in the ontology system. Before accessing or searching the details user should have the account in that otherwise they should register first. At the very least, you need to provide an email address, username, password, display name, and whatever profile fields you have set to required. The display name is what will be used when the system needs to display the proper name of the user.

Server Start Up And Upload File

The user can start up the server after cloud environment is opened. Then the user can upload the file to the cloud.

Secure De Duplication System

To support authorized de duplication the tag of a file F will be determined by the file F and the privilege. To show the difference with traditional notation of tag, we call is file token instead. To support authorized access a secret key KP will be bounded with a privilege p to generate a file Token. De duplication exploits identical content, while encryption attempts to make all content appear random; the same content encrypted with two different keys results in very different cipher text. Thus, combining the space efficiency of de duplication with the secrecy aspects of encryption is problematic.

Download File

After the cloud storage, the user can download the file based on key or token.Once the key request was received, the sender can send the key or he can decline it. With this key and request id which was generated at the time of sending key request the receiver can decrypt the message.

IX. CONCLUSION

In this paper, the notion of authorized data de duplication was proposed to protect the data security by including differential privileges of users in the duplicate check. We also presented several new de duplication constructions supporting authorized duplicate check in hybrid cloud architecture, in which the duplicate-check tokens of files are generated by the private cloud server with private keys. Security analysis demonstrates that our schemes are secure in terms of insider and outsider attacks specified in the proposed security model. As a proof of concept, we implemented a prototype of our proposed authorized duplicate check scheme and conduct test bed experiments on our prototype. We showed that our authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and network transfer.

FUTURE ENHANCEMENT

Finally, we believe that cloud data storage security is still full of challenges and of paramount importance, and many research problems remain to be identified.

REFERENCES

[1] 1.M. Bellare, S. Keelveedhi, and T. Ristenpart. Dupless: Serveraided encryption for deduplicated storage. In USENIX Security Symposium, 2013.

[2] 2.M. Bellare, S. Keelveedhi, and T. Ristenpart. Message-locked encryption and secure deduplication. In EUROCRYPT, pages 296–312, 2013.

[3] 3.M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. J. Cryptology, 22(1):1–61, 2009.

[4] 4.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.

[5] 5.S. Bugiel, S. Nurnberger, A. Sadeghi, and T. Schneider. Twin clouds: An architecture for secure cloud computing. In Workshop on Cryptography and Security in Clouds (WCSC 2011), 2011.

[6] 6.P. Anderson and L. Zhang. Fast and secure laptop backups with encrypted de-duplication. In Proc. of USENIX LISA, 2010.

[7] 7.M. Bellare, S. Keelveedhi, and T. Ristenpart. Dupless: Serveraided encryption for deduplicated storage. In USENIX Security Symposium, 2013.

[8] 8..M. Bellare, S. Keelveedhi, and T. Ristenpart. Message-locked encryption and secure deduplication. In EUROCRYPT, pages 296– 312, 2013.

[9] 9.M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. J. Cryptology, 22(1):1–61, 2009.