ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

IJARCCE

nCORETech

LBS College of Engineering, Kasaragod

Vol. 5, Special Issue 1, Feburary 2016

SecDedup: A Secure Way for Maximizing Space Savings in Cloud Storage

Bushra K.R.[1], **Sindhu S**[2]

M.Tech, Computer Science and Engineering, NSS College of Engineering, Palakkad, Kerala, India[1]

Assoc. Professor, Computer Science and Engineering, NSS College of Engineering, Palakkad, Kerala, India[2]

Abstract: Security and privacy are important concerns for the public cloud environments. One of the critical challenges of cloud storage services is the increasing volume of data. This burdens the cloud storage which is not infinitely large. A widely used method as a solution for this is, data deduplication which saves storage space and upload bandwidth. The commonly used technique for deduplication is convergent encryption. The notion of authorized data deduplication is not so popular in the cloud environment. Hybrid architecture is used here to provide authorized data deduplication. Further, this method is enhanced by providing block-level deduplication.

Keywords: Deduplication, Hybrid Cloud, Merkle-Hash tree, Proof of Ownership.

I.INTRODUCTION

Cloud computing refers to applications and services that run on a distributed network using virtualized resources and accessed by common Internet protocols and networking standards. The users of cloud storage service can upload and download their own data anytime and anywhere via any devices by connecting to the cloud. The fast growth of data volumes burdens the cloud storage which is finite.

Data deduplication is a technique used to look for the redundant files or data blocks and just stores one copy of them. Owing to this technology, the cloud storage space can be used efficiently when there are some files that were stored frequently. Deduplication can be of many types of which important are file level deduplication and block level deduplication. File level deduplication is a method of keeping a single copy of a file and eliminating the redundant ones. In block-level deduplication a file is divided into various blocks and deduplication is applied to these blocks of data. The blocks can be fixed sized or variable sized.

There are three factors on security of cloud computing: Data confidentiality, Integrity, and Availability. On the storage service, users worry about the personal privacy, which is data confidentiality. No one would like to disclose their sensitive data to others. On most of cloud storage services, the cloud server can obtain and control all the users' data as those data are stored in the cloud storage in unencrypted format. In order to protect the privacy, the cloud users can use some cryptographic techniques to encrypt files before they are uploaded to the cloud storage. Consequently, the users can keep their data secret, and the cloud server cannot know the information of each ciphertext. When the users want to access their data, they just need to download the corresponding ciphertexts from cloud storage and decrypt them. The goal of encryption is to keep the information private and make the ciphertext indistinguishable from a random value. It is not easy to verify whether two given ciphertexts were encrypted from the same plaintext or not. In contrast, the goal of data deduplication is to reduce the requirement of storage space by sharing only one copy of the same plaintext or information. Therefore, it is conflicting between data deduplication and confidentiality. If data deduplication on ciphertexts is possible, the cloud server must be able to identify all of the ciphertexts of the same plaintext. In such cases, the encryption may not be semantically secure, because adversaries can get non-trivial secret information from the ciphertexts without the decryption keys.

To prevent unauthorized access, a secure proof of ownership (POW) protocol is needed to prove that the user indeed owns the same file when a duplicate is found. After the proof, the users with the same file will be provided a pointer from the server and they need not upload the same file. A user can download the encrypted file with the pointer from the server, which can be decrypted by the corresponding data owners with their convergent keys.
The authorized deduplication system consists of a token generator and a public cloud. Each user can perform only a set of operations allowed according to the set of privileges assigned to the user by the token generator. Those set of privileges specify which kind of users can perform the duplicate check and access the files. Block level deduplication is performed.

II. RELATED WORKS

With the advent of cloud computing, secure data deduplication techniques have got wide popularity. Following are the different methods which are used for secure data deduplication in cloud storage.

Harnik et. al. [2] mentions various side channel attacks that are possible in deduplication. These attacks are mainly: (i) Confirmation of file (ii) Learning the remaining parts of the file (iii) Covert channels attacks. They also propose various methods to overcome these attacks. But these methods do not provide sufficient security when inter-user deduplication is enabled. To deal with the security issues introduced by a client and inter-user deduplication, a method called randomized solution is proposed. This solution sets a random threshold for each and every file in the storage system which corresponds to the number of different clients who already have uploaded the same file. Each time a client uploads a file for the first time; a counter is created associating each file and it is set to 1 initially. As long as the counter is under the threshold, deduplication is not needed at all. When the counter reaches the threshold, the inter-user deduplication is activated and applied to all the subsequent uploads of that file. This solution is reliable only up to when the threshold is not reached. As soon as the system applies deduplication to a file, it becomes vulnerable to many attacks. Moreover, the number of copies is restricted to the threshold and not to 1. This saves storage space but not to the best.

In [3] Choi and Lee enhance the method used in [1]. This paper revisits Harnik's solution and proposes an improved solution for cross-user source-based deduplication with dramatically reduced risk of information leakage. The authors prove mathematically that this paper provides security against side channel information leakage.

The paper [4] copes with the security vulnerabilities of convergent encryption and proposes ClouDedup, which utilizes the advantages of both data deduplication and convergent encryption. The security of ClouDedup depends on its architecture in which, a metadata manager and an additional server are defined in addition to the basic storage provider. The server adds an additional encryption layer to prevent major attacks against convergent encryption. The metadata manager is responsible for the key management tasks. Therefore, the underlying deduplication is performed at block-level and an efficient key management mechanism is defined to avoid users to store one key per block. This method is transparent to storage provider.

A message in the proposed hybrid data deduplication mechanism [5] consists of a triple of blocks, check block, enabling block and cipher block. The check block can be used to check the repetition of encrypted files. A session key (an AES key) which is used to encrypt data is stored in the enabling block. The cipher block contains the encrypted data that are encrypted with the session key. The system model of this mechanism contains two types of cloud storage. One is an un-encrypted area where plaintexts are kept. The other is an encrypted area where encrypted data is kept. Data is encrypted with a user's key. The most duplicate files which were frequently uploaded by different users are often commonly used files such as multimedia files, software, and so on. If an uploaded ciphertext contains a commonly-used file, the cloud server will perform the proposed ciphertext deduplication mechanism on it. Otherwise, the server will store the received ciphertext and none can learn anything about the received ciphertexts. But, here bandwidth consumption is more.

The paper [6] introduces a new cryptographic method for secure Proof of Ownership (POW). It is based on the combined use of convergent encryption and the Merkle-hash Tree. This helps in enhancing security of data in cloud storage. It provides dynamic sharing between users and ensuring efficient data deduplication. The idea consists in using the Merkle-based Tree over encrypted data, for deriving a unique identifier of uploaded data. This identifier checks the availability of the redundant data in remote cloud servers. It is used to ensure access control in dynamic sharing of data. This paper deals with COF and LRI attacks. In this approach, each user has to store his private keys in the cloud.

A new construction called Dekey, which guarantees efficient and reliable convergent key management on both user and cloud storage sides is proposed in [7]. The idea is to apply deduplication to the Convergent keys and leverage secret sharing techniques. This method does not solve the vulnerabilities of traditional convergent encryption.

Jin Li, Yan Kit Li, X. Chen, Patrick P.C. Lee, and W. Lou [8] propose a new system based on hybrid cloud architecture. The data owners only store their data in the public cloud whereas the data operation is handled by private cloud. An advanced scheme is proposed to support stronger security by encrypting the file with privilege keys. The users without sufficient privileges cannot perform the duplicate check. Description of Proof of Ownership protocol is vague. File level deduplication is performed here.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

IJARCCE

nCORETech

LBS College of Engineering, Kasaragod

Vol. 5, Special Issue 1, Feburary 2016

III. PRELIMINARIES

Convergent Encryption: The basic idea of convergent encryption is to derive the encryption key from the hash of the plaintext. Message M can be encrypted such that encryption key is K = H (M), where H is a cryptographic hash function. Ciphertext, C = E (K, M) = E (H (M), M), where E is a block cipher.

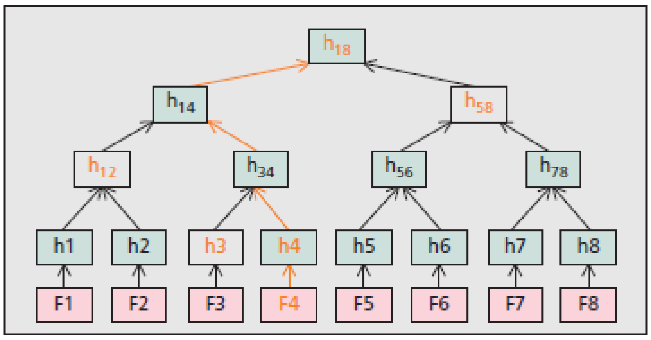

Fig1. Merkle Hash tree

Proof of Ownership: Proof of ownership helps users to prove their ownership of data copies to the storage server. POW is employed as an interactive algorithm run by a prover (user) and a verifier (storage server).

The Merkle-tree proof protocol: A verifier has the root value of a Merkle tree MTh;b(X) and the number of leaves in the tree. The prover claims to be the owner of the file. The verifier simply chooses some number of leaf indexes and asks the prover for the value of the corresponding leaves. The prover replies with these leaves and with a valid sibling path for each one of them and the verifier accepts only if all these sibling paths are valid.

IV.SYSTEM DETAILS

This is a novel architecture for data deduplication in cloud computing which consists of twin clouds- public cloud and a token generator. The setting of interest for this system is an enterprise network, consisting of a group of clients who store data with deduplication. There are three entities defined in the system:

- Data Users: Outsource data to the storage provider and access the data later. In a storage system that supports deduplication, the user uploads unique data but cannot upload any duplicate data, which may be possessed by the same user or different users; to save the upload bandwidth. In this system, each user is issued a set of privileges. Each file is encrypted using a convergent encryption key and privilege key for each user.

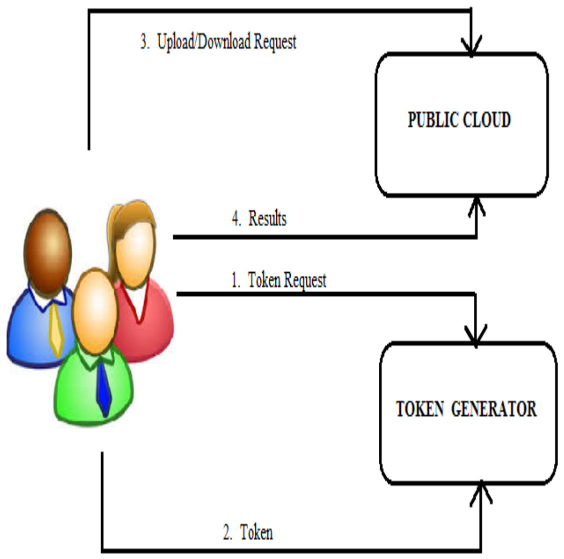

Fig2. Authorized deduplication system architecture

- Token generator: Entity for managing the privilege keys issued for the users. It also provides file token to users.
- S-CSP in Public cloud: Provides data outsourcing service and stores data on behalf of users. This entity eliminates storage of redundant copies of data by deduplication.

The new system supports both differential authorization and authorized duplicate check. For the security of file token, two aspects are defined: Unforgeability and Indistinguishability of file token and Data Confidentiality. The access rights are defined based on a set of privileges. Privileges can be such that as those based on the time-period or on role of the authority. Each file is associated with some file tokens. A token is a file tag with specified privileges. A user computes and sends duplicate check tokens to the public cloud for authorized duplicate check.

A differential authorization duplicate check is supported by the system. Public cloud is used to store data and deduplication technique is applied here. Each file uploaded is associated with a set of privileges and only the users with the specified privileges can access the file. The Token generator handles privileges of users and generation of file tokens. File tokens uniquely distinguish a file with specified privileges. The figure above shows the architecture of the system.

Two levels of deduplication are implemented. Block-level deduplication and file level deduplication. To overcome the side channel attacks and the users to prove the ownership of data copies proof of ownership is used. To enhance security, two levels of security can be implemented while accessing the token generator– Password based and OTP based. Hashing can be done based on multiple hash functions for security of predictable messages. To avoid the security vulnerabilities of traditional convergent encryption, multiple hashing is done.

To enable block-level deduplication, a file is divided into a number of blocks and these blocks are checked for duplicates. To upload a file, a user first checks for file level duplicates. If a duplicate file is found, then all its blocks must be duplicates as well; else, the user further checks for block-level duplicates and identifies the unique blocks.

Here, a new cryptographic method is used for secure Proof of Ownership; that is based on Merkle-Hash Tree, for enhancing data security in cloud storage systems and avoiding side channel attacks. The Merkle-based Tree is used over encrypted data, in order to derive a unique identifier of uploaded data. The Merkle tree proof protocol checks whether the prover has the original data file. The verifier chooses a number of leaf indexes and asks the prover to provide the corresponding leaves. As such, the prover has to send these leaves with a sibling valid path.

A. File Uploading

a. Token generator will find the corresponding privileges from stored table.
b. User computes file tag and sends this to token generator.
c. The token generator verifies the credentials and returns the user with a token.
d. The user interacts with storage provider in the public cloud and sends the file token.
e. If a duplicate file is found, the Merkle hash index of the file is calculated. If the index already exists in the database, user runs POW protocol with the storage provider.
i. The proof is returned to the token generator and the token generator will update the list of data owners.
f. If no duplicate is found, the file is divided into blocks to perform block level deduplication. The following steps will be taken.
  i. Steps (a-d) take place.
  ii. If a duplicate block is found, the Merkle hash index of the file is calculated. If the index already exists in the database, user runs POW protocol with the storage provider.
  iii. The proof is returned to the token generator and the token generator will update the list of data owners.
g. The convergent key is calculated by the user.
h. The user encrypts the plaintext with convergent key.
i. Then the user uploads the ciphertext after verification by the cloud servers.

B. File Retrieving

a. The user sends request to the storage provider.
b. The storage provider checks whether the user can download the file/block.
c. If ok, the storage provider sends the ciphertext to the user.
d. Else, sends a signal to indicate download failure.

V. IMPLEMENTATION

The system is implemented in .Net framework. Amazon Web Services is used as the Public cloud. AWS S3 is used to store files. AWS RDS is used for data management.

In the implementation of user process, following function calls and algorithms are used:

- File Tag (File) – SHA-1 Hash of the file.
- TokenReq (Tag, UserID) — Requests the Private Server for File Token with the File Tag and User ID.
- DupCheckReq (Token)—Requests the Storage Server for Duplicate Check of the File by sending the file token received from private server.
- FileEncrypt (File) — File is encrypted with Convergent Encryption using 256-bit AES algorithm in cipher block chaining (CBC) mode.
- ○ Convergent key— SHA-256 Hash of the file.
- FileUploadReq (FileID, File, Token) — Uploads the File to the Storage Server if the file is Unique and updates the File Token stored.

In the implementation of private server, following function calls and algorithms are used:

- TokenGen (Tag, UserID)—Loads the privilege keys associated with user and generate the token with HMAC-SHA-1 algorithm.

In the implementation of public server, following function calls and algorithms are used:

- DupCheck (Token)—Checks the file for duplicates.
- FileStore (FileID, File, Token) — Stores the File on Disk.

Variable block level deduplications can be done using Rolling hash functions. To avoid hash collisions a unique identifier is calculated for files or blocks having same tokens. The unique identifier is a Merkle identifier created by the user and stored by token generator. The identifier is created by dividing the file into a fixed number of blocks and creating a tree upon the blocks.

VII. CONCLUSION

In this paper, a method for authorized data deduplication is proposed. This method overcomes the security vulnerabilities of traditional Convergent Encryption. It also provides file level as well as block level deduplication in cloud storage to save storage space and bandwidth. Moreover, it is based on Merkle-hash tree based method to avoid side channel attacks and hash collisions. The research can be extended by using Homomorphic encryption which is widely used for cloud security. Also, deduplication technique can be performed upon image and video files.

REFERNCES

[1] Jin Li Yan Kit Li, Xiaofeng Chen, Patrick P.C. Lee, and Wenjing Lou, "A Hybrid Cloud Approach for Secure Authorized Deduplication", IEEE Transactions On Parallel And Distributed Systems, Vol. 26, No. 5, May 2015.
[2] Danny Harnik, Benny Pinkas and Alexandra Shulman-Peleg, "Side channels in cloud services: Deduplication in cloud storage", Security Privacy, IEEE, 8(6):40-47, 2010.
[3] Seungkwang Lee, Dooho Choi, "Privacy- Preserving Cross-User Source-Based Data Deduplication in Cloud Storage ", ICTC, 2012.
[4] Puzio P., Molva, R., Onen, M., Loureiro, S., "ClouDedup: Secure Deduplication with Encrypted Data for Cloud Storage ", IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom), 2013. pp. 363 - 370.
[5] Chun-I Fan, Shi-Yuan Huang, Wen-Che Hsu, "Hybrid Data Deduplication in Cloud Environment ",International Conference on Information Security and Intelligence Control (ISIC), 2012 .pp. 174-177.
[6] Nesrine Kaaniche, Maryline Laurent, "A Secure Client Side Deduplication Scheme in Cloud Storage Environments ", IEEE Transactions on .Mobility and Security (NTMS) in Cloud Computing, April.2014.pp.1-8.
[7] S.Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems," in Proc. ACM Conf. Comput Commun. . Security, 2011, pp. 491–500.
[8] AWS Toolkit for Visual Studio: User Guide, Version v1.30, 2016.
[9] http://aws.amazon.com/documentation.
[10] Barrie Sosinsky, "Cloud Computing Bible", Wiley Publishing, Inc., 2011.