IISRC International Industrial Scientific Research Corporation
International Conference on Industrial Scientific Research Engineering
www.worldconferences.org

Secure Authorized Deduplication using Hybrid Cloud Approach

SABDUL AZAM[1], S.G.NAWAZ[2], M.HARATHI[3]
[1]PG Scholar, Dept of CSE Sri Krishna Devaraya Engineering College, Gooty, AP, India,
E-mail: azam.1947@gmail.com.
[2]HOD, Dept of CSE Sri Krishna Devaraya Engineering College, Gooty, AP, India.
[3]Associate Professor, Dept of CSE Sri Krishna Devaraya Engineering College, Gooty, AP, India.

Abstract: Data deduplication is an important technique for eliminating redundant data. Instead of taking no. of same files, it store only single copy of file. In most organizations, storage system contains many pieces of duplicate data. For example, the same file may be saved in several different places by different users. Deduplication eliminates these extra copies by saving just one copy of the data and replacing the other copies with pointers that lead back to the original copy. It is data compression technique for improve the bandwidth efficiency and storage utilization. Data deduplication most widely used in cloud computing. It make data management scalable and storage problem in cloud computing. Data deduplication protects the confidentiality of sensitive data. Data deduplication work with convergent encryption technique to encrypt the data before uploading. . Companies frequently use deduplication in backup and disaster recovery applications. In this paper we attempt authorized deduplication check, combine with convergent encryption for providing security to sensitive data using hybrid cloud computing.

Keywords: Deduplication, Authorized Duplicate Check, Confidentiality, Hybrid Cloud, Covergent Encryption.

I. INTRODUCTION

Cloud computing technique which is most widely used todays. In that, computing is done over the large communication network like Internet. It is an important solution for business storage in low cost. Cloud computing provide vast storage in all sector like government, enterprise, also for storing our personal data on cloud. Without background implementation details, Platform user can access and share different resources on cloud. The most important problem in cloud computing is that large amount of storage space and security issues. One critical challenge of cloud storage to management of ever-increasing volume of data. To improve scalability, storage problem data deduplication is most important technique and has attracted more attension recently. It is an important technique for data compression, It simply avoid the duplicate copies of data and store single copy of data. Data deduplication take place in either block level or file level. In file level approach duplicate files are eliminate, and in block level approach duplicate blocks of data that occur in non-identical files.

Deduplication reduce the storage needs by upto 90-95% for backup application, 68% in standard file system. Important issues in data deduplication that security and privacy to protect the data from insider or outsider attack. For data confidentiality, encryption is used by different user for encrypt their files or data, using a secrete key user perform encryption and decryption operation. For uploading file to cloud user first generate convergent key, encryption of file then load file to the cloud. To prevent unauthorized access proof of ownership protocol is used to provide proof that the user indeed owns the same file when deduplication found. After the proof, server provides a pointer to subsequent user for accessing same file without needing to upload same file. When user want to download file he simply download encrypted file from cloud and decrypt this file using convergent key.

II. CONVERGENT ENCRYPTION

Convergent encryption is used to encrypt and decrypt file. User can derives the convergent key from each original data copy, then using that key encrypt data file. Also user derives tag for data copy to check duplicate data. If tags are same then both files are same. Both convergent key and tag are independently derives. Convergent encryption, also known as content hash keying, is used to produces identical ciphertext from identical plaintext files. The simplest implementation of convergent encryption can be defined as: Alice derives the encryption key from her file F such that K = H(F), where H is a cryptographic hash function. Convergent encryption scheme can be defined with four primitive functions: [1] KeyGenCE(M) -> K is the key generation algorithm that maps a data copy M to a convergent key K; [2] EncCE(K, M) ->C is the symmetric encryption algorithm that takes both the convergent key K and the data copy M as inputs and then outputs a ciphertext C;

[3] DecCE (K, C) ->M is the decryption algorithm that takes both the ciphertext C and the convergent key K as inputs and then outputs the original data copy M and TagGen(M) -> T (M) is the tag generation algorithm that maps the original data copy M and outputs a tag T(M).

III. PROOF OF OWNERSHIP

Proof of ownership (PoW) is a protocol enables users to prove their ownership of data copies to the storage server. PoW is implemented as an interactive algorithm run by user and storage server act as prover and verifier. The verifier derives a short value ϕ(M)from a data copy M . To prove the ownership of the data copy M , the prover needs to send ϕ to the verifier such that ϕ= ϕ(M). The formal security definition for PoW roughly follows the threat model in a content distribution network, where an attacker does not know the entire file, but has accomplices who have the file. Proof of-ownership is specified by a summary function S(.) (Which could be randomized and takes the input file F and a security parameter), and an interactive two-party protocol ||(P V). To solve the problem of using a small hash value as a proxy for the entire file, we want to design a solution where a client proves to the server that it indeed has the file. We call a proof mechanism that prevents such leakage amplification a proof of ownership (PoW).

Public Hash Function: To support cross-user deduplication, all users must use the same procedure for identifying duplicate files. Hence this procedure must be public i.e. each user implement it, which means that a determined attacker can learn it.

IV. SYSTEM MODEL

A. Hybrid Architecture for Secure Deduplication

At a high level, our setting of interest is an enterprise network, consisting of a group of affiliated clients (for example, employees of a company) who will use the S-CSP and store data with deduplication technique. In this setting, deduplication can be frequently used in these settings for data backup and disaster recovery applications while greatly reducing storage space. Such systems are widespread and are often more suitable to user file backup and synchronization applications than richer storage abstractions. There are three entities defined in our system, that is, users, private cloud and S-CSP in public cloud as shown in Fig.1. The S-CSP performs deduplication by checking if the contents of two files are the same and stores only one of them. The access right to a file is defined based on a set of privileges. The exact definition of a privilege varies across applications. For example, we may define a role based privilege according to job positions (e.g., Director, Project Lead, and Engineer), or we may define a time-based privilege that specifies a valid time period (e.g., 2014-01-01 to 2014-01-31) within which a file can be accessed. A user, say Alice, may be assigned two privileges "Director" and "access right valid on 2014-01-01", so that she can access any file whose access role is "Director" and accessible time period covers 2014-01-01.

Each privilege is represented in the form of a short message called token. Each file is associated with some file tokens, which denote the tag with specified privileges (see the definition of a tag in Section 2). A user computes and sends duplicate-check tokens to the public cloud for authorized duplicate check. Users have access to the private cloud server, a semi trusted third party which will aid in performing deduplicable encryption by generating file tokens for the requesting users. We will explain further the role of the private cloud server below. Users are also provisioned 4 with per-user encryption keys and credentials (e.g., user certificates). In this paper, we will only consider the file level deduplication for simplicity. In another word, we refer a data copy to be a whole file and file-level deduplication which eliminates the storage of any redundant files. Actually, block-level deduplication can be easily deduced from file-level deduplication, which is similar to. Specifically, to upload a file, a user first performs the file-level duplicate check. If the file is a duplicate, then all its blocks must be duplicates as well; otherwise, the user further performs the block-level duplicate check and identifies the unique blocks to be uploaded. Each data copy (i.e., a file or a block).

V. DESIGN GOALS

In this paper, we address the problem of privacy preserving deduplication in cloud computing and propose a new deduplication system supporting for:

A. Differential Authorization

Each authorized user is able to access its individual token of his file to perform duplicate check based on authority. Under this assumption, any user cannot generate a token for duplicate check out of his access or without the aid from the private cloud server.

B. Authorized Duplicate Check

Authorized user is able to access his/her own token from private cloud, while the public cloud performs duplicate check directly and tells the user if there is any duplicate. The security requirements considered in this paper lie in two folds, including the security of file token and security of data files. For the security of file token, two aspects are defined as unforgeability and indistinguishability of file token. The details are given below.

C. Unforgeability Of File Token/Duplicate-Check Token

User make registration in private cloud for generating file token. Using respective file token he/she upload or download files on public cloud. The users are not allowed to collude with the public cloud server to break the unforgeability of file tokens. In our system, the S-

CSP is honest but curious and will honestly perform the duplicate check upon receiving the duplicate request from users. The duplicate check token of users should be issued from the private cloud server in our scheme.

VI. PROPOSED SYSTEM

In our system we implement a project that includes the public cloud and the private cloud and also the hybrid cloud which is a combination of the both public cloud and private cloud. In general by if we used the public cloud we can't provide the security to our private data and hence our private data will be loss. So that we have to provide the security to our data for that we make a use of private cloud also. When we use a private clouds the greater security can be provided. In this system we also provides the data deduplication. Which is used to avoid the duplicate copies of data. User can upload and download the files from public cloud but private cloud provides the security for that data. That means only the authorized person can upload and download the files from the public cloud. For that user generates the key and stored that key onto the private cloud. At the time of downloading user request to the private cloud for key and then access that Particular file.

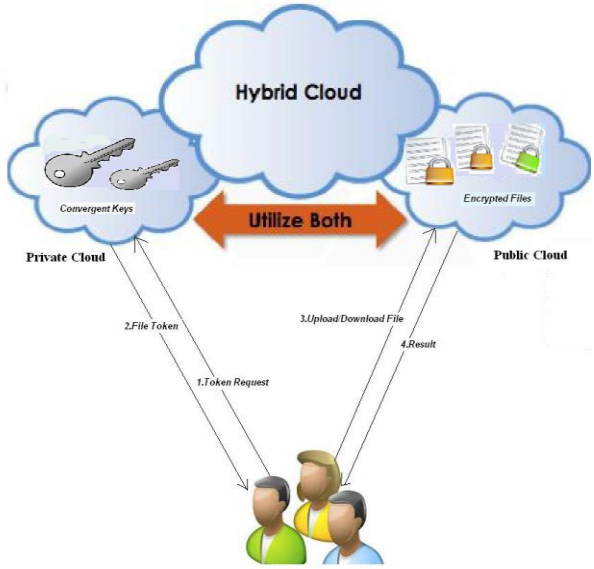

Fig.1. Architecture of Authorized Deduplication.

A. Roles Of Entities

S-CSP: The purpose of this entity to work as a data storage service in public cloud. On the half of the user S-CSP stores the data. The S-CSP eliminate the duplicate data using deduplication and keep the unique data as it is. SSCP entity is used to reduce the storage cost. S-CSP han abundant storage capacity and computational power. When user send respective token for accessing his file from public cloud S-CSP matches this token with internally if it matched then an then only he send the file or ciphertext Cf with token, otherwise he send abort signal to user. After receiving file user use convergent key KF to decrypt the file.

Data User: A user is entities that want to access the data or files from S-SCP. Users generate the key and store that key in private cloud. In storage system supporting deduplication, The user only upload unique data but do not upload any duplicate data to save the upload bandwidth, which may be owned by the same user or different users. Each file is protected by convergent encryption key and can access by only authorized person. In our system user must need to register in private cloud for storing token with respective file which are store on public cloud. When he want to access that file he access respective token from private cloud and then access his files from public cloud. Token consist of file content F and convergent key KF.

Private Cloud: In general for providing more security user can use the private cloud instead of public cloud. User stores the generated key in private cloud. At the time of downloading system ask the key to download the file. User can not store the secrete key internally. For providing proper protection to key we use private cloud. Private cloud only stores the convergent key with respective file. When user want to access the key he first check authority of user then an then provide key.

Public Cloud: Public cloud entity is used for the storage purpose. User uploads the files in public cloud. Public cloud is similar as S-CSP. When the user wants to download the files from public cloud, it will be ask the key which is generated or stored in private cloud. When the users key is match with files key at that time user can download the file, without key user can not access the file. Only authorized user can access the file. In public cloud all files are stored in encrypted format. If any chance unauthorized person hack our file, but without the secrete or convergent key he doesn't access original file. On public cloud there are lots of files are store each user access its respective file if its token matches with S-CSP server token.

VII. IMPLEMENTATION

We implement system with data deduplication, in which we model three entities as separate programs. A Client program is used to model the data users to carry out the file upload/download process. A Private Server program is used to model the private cloud which manages the private keys and handles the file token computation. A Storage Server program is used to model the S-CSP which stores and deduplicates files. Followings are function calls used in system:

- **FileTag(File):** It computes SHA-1 hash of the File as File Tag.
- **DupCheckReq(Token):** It requests the Storage Server for Duplicate Check of the file.
- **FileEncrypt(File):** It encrypts the File with Convergent Encryption using 256-bit AES algorithm in cipher block chaining (CBC) mode, where the convergent key is from SHA-256 Hashing of the file.

- **FileUploadReq (FileID, File, Token):** It uploads the File Data to the Storage Server if the file is Unique and¬ updates the File Token stored.
- **FileStore (FileID, File, Token):** It stores the File on Disk and updates the Mapping.

IX. CONCLUSION

In this paper, the notion of authorized data deduplication was proposed to protect the data security by including differential privileges of users in the duplicate check. We also presented several new deduplication 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 testbed experiments on our prototype. We showed that our authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and network transfer. In public cloud our data are securely store in encrypted format, and also in private cloud our key is store with respective file. There is no need to user remember the key. So without key anyone can not access our file or data from public cloud.

X. REFERENCES

[1] OpenSSL Project. http://www.openssl.org/.
[2] P. Anderson and L. Zhang. Fast and secure laptop backups with encrypted de-duplication. In Proc. of USENIX LISA, 2010.
[3] M. Bellare, S. Keelveedhi, and T. Ristenpart. Dupless: Serveraided encryption for deduplicated storage. In USENIX Security Symposium, 2013.
[4] M. Bellare, S. Keelveedhi, and T. Ristenpart. Message-locked encryption and secure deduplication. In EUROCRYPT, pages 296– 312, 2013.
[5] M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. J. Cryptology, 22(1):1–61, 2009.