Deduplication on Cloud using Hybrid cloud approach with secured authorized duplicate check Mechanism

[1]Ms.R.Rajamuthupetchi
Assistant Professor
Electronics and Communication Engineering
PSR Engineering College
Sivakasi, India

***ABSTRUCT*— Data deduplication is one of important data compression techniques for eliminating duplicate copies of repeating data and has been widely used in cloud storage in order to minimize the amount of storage space and save bandwidth. To protect the confidentiality of important data while supporting deduplication, the convergent encryption technique has been proposed to encrypt the data before outsourcing. Along with the data, the privilege level of the user is also checked in order to assure whether he is an authorised user or not. As a proof of concept, we implemented a prototype of our proposed authorized duplicate check scheme and conduct tested experimented using our prototype. This paper tries to minimize the data duplication that occurs in hybrid cloud storage by using various techniques such as Converegent encryption, Proof of Ownership (POW).**

Index Terms— ***Deduplication, authorized duplicate check, confidentiality, hybrid cloud***

I.INDRODUCTION

Deduplication can take place at the file level or either block level. For the 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 there are several advantages of data deduplication security and privacy concerns arise as users' sensitive data are susceptible to both insider's and outsider's attacks. Encryption techniques which were used traditionally were not compatible with data deduplication while providing data confidentiality. 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. Whenever the key is generated, users retain the keys and send the cipher text to the cloud. In order 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. Hence convergent encryption allows the cloud to perform deduplication on the cipher texts and the proof of ownership prevents the unauthorized user to access the file. Traditional deduplication systems based on convergent encryption, although providing confidentiality to some extent; do not support the duplicate check with differential privileges

II.PRELIMINARIES

In this section, we define the notations used in this paper, review some secure primitives used in our secure duplication.

1.Symmetric encryption

Symmetric encryption uses a common secret key*k*toencrypt and decrypt information. A symmetric encryption scheme consists of three Primitive functions:

KeyGenSE $(1^\lambda) \rightarrow \kappa$ is the key generation algorithm that generates κ using security parameter 1; DecSE $(\kappa, C) \rightarrow M$ is the symmetric decryption algorithm that takes the secret κ and ciphertext$C$ and then outputs the original message M.

EncSE $(\kappa, M) \rightarrow C$ is the symmetric encryption algorithm that takes the secret κ and message M and then outputs the ciphertext C.

1.1Convergent encryption

Convergen tencryption provides data confidentiality in deduplication. A data owner derives a convergent key from each original data copy and encrypts the data copy with the convergent key.

KeyGenCE $(M) \rightarrow K$ is the key generation algorithm that maps a data copy M to a convergent key K

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*.

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*;

TagGen (*M*) → *T* (*M*) is the tag generation algorithm that maps the original data copy *M* and outputs a tag *T* (*M*).

DecSE (*κ, C*) → *M* is the symmetric decryption algorithm that takes the secret *κ* and ciphertext*C* and then outputs the original message *M*.

2.***Proof of ownership***

The notion of proof of ownership (POW) [2] enables users to prove their ownership of data copies to the storage server. However PoW is implemented as an interactive algorithm (denoted by PoW) run by a proverb (i.e., user) and a verifier (i.e.storage server).

3.***Identification Protocol***

An identification protocol can be described with two phases: Proof and Verify. In the stage of Proof, aprover/user *U* can demonstrate his identity to a verifier by performing some identification proof related to his identity. The input of the prover/user is his private key *skU* that is sensitive information such as private key of a public key in his certificate or credit card number etc. that he would not like to share with the other users.

III.SYSTEM MODEL

3.1Hybrid 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. 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. 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. Role of the private cloud server will be explained in the paper. block-level deduplication can be easily deduced from file-level deduplication. 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.

1.S-CSP

This is an entity that provides a data storage service in public cloud. The S-CSP provides the data outsourcing service and stores data on behalf of the users.

2.Data Users

A user is an entity that wants to outsource data storage to the S-CSP and access the data later. In a storage system supporting deduplication, the user only u ploads unique data but does not upload any duplicate data

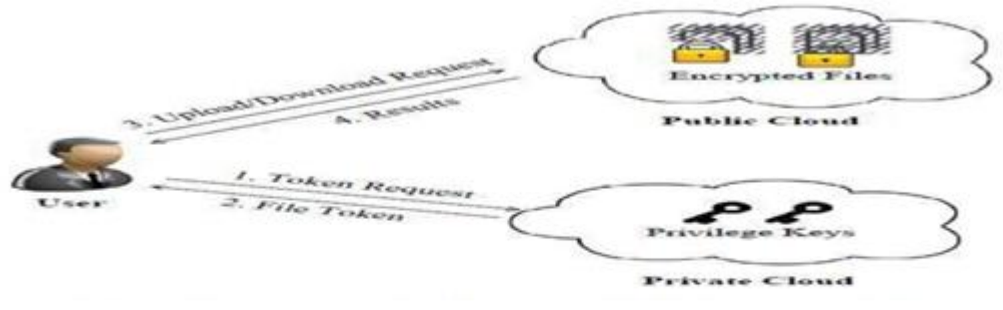

Fig1. Architecture for authorized deduplication

3.Private Cloud

Compared with the traditional deduplication architecture in cloud computing, this is a new entity introduced for facilitating user's secure usage of cloud service. Private Keys are managed by private cloud in order to give them privileges as per their designation.

4.Design Goals

We have proposed a new deduplication system for the following:

5.Differential Authorization

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

6.Authorized Duplicate Check

Authorized user is able to use his/her individual private keys to generate query for certain file and the privileges he/she owned with the help of private cloud, while the public cloud performs duplicate check directly and tells the user if there is any duplicate.

7.Unforgeability of file token/duplicate-check token

Unauthorized users without appropriate privileges or file should be prevented from getting or generating the file tokens for duplicate check of any file stored at the S-CSP. The duplicate check token of users should be issued from the private cloud server in our scheme

IV.SECURE DEDUPLICATION SYSTEMS

4.1Problem Statement

Existing deduplication system cannot prevent the privilege private key sharing among users. The users will be issued the same private key for the same privilege in the construction. As a result, the users may collude and generate privilege private keys for a new privilege set $P*$ that does not belong to any of the colluded user. For example, a user with privilege set $PU1$may collude with another user with privilege set $PU2$to get a privilege set

$$P*=PU1 \cup PU2.$$

The construction is inherently subject to brute-force attacks that can recover files falling into a known set. That is, the deduplication system cannot protect the security of predictable files. One of critical reasons is that the traditional convergent encryption system can only protect the semantic security of unpredictable files.

4.2Proposed System

The users cannot share these private keys of privileges in this proposed construction, which means that it can prevent the privilege key sharing among users in the above straightforward construction. To get a file token, the user needs to send a request to the private cloud server. The intuition of this construction can be described as follows. To perform the duplicate check for some file, the user needs to get the file token from the private cloud server. The private cloud server will also check the user's identity before issuing the corresponding file token to the user. The authorized duplicate check for this file can be performed by the user with the public cloud before uploading this file. Based on the results of duplicate check, the user either uploads this file orruns PoW.

1.

1.System Setup

An identification protocol $\prod$ = (Proof,Verify) is also defined, where Proof and Verify are the proof and verification algorithm respectively. Furthermore, each user U is assumed to have a secret key skU to perform the identification with servers. Assume that user U has the privilege set PU. It also initializes a PoW protocol POW for the file ownership proof. The private cloud server will maintain a table which stores each user's public information pkU and its corresponding privilege set PU. The file storage system for the storage server is set to be

2.File Uploading

Suppose that a data owner wants to upload and share a file$F$ with users whose privilege belongs to the setPF= $\{pj\}$. The data owner needs interact with the private cloud before performing duplicate check with the S-CSP. Data owner performs an identification to prove its identity with private key skU. If it is passed, the private cloud server will find the corresponding privileges PU of the user from its stored table list. The user computes and sends the file tag ϕF = TagGen (F) to the private cloud server, who will return $\phi' F, p$ = TagGen ($\phi F, kpr$) g back to the user for all pr satisfying R (p, pr) = 1 and $p \in PU$. Then, the user will interact and send the file token $\{\phi' Fop\}$ to the S-CSP.

If a file duplicate is found, the user needs to run the POW protocol POW with the S-CSP to prove the file ownership. If the proof is passed, the user will be provided a pointer for the file. Furthermore, a proof from the S-CSP will be returned, which could be a signature on $\{\phi'$ F, $pr\}$, PKU and a time stamp. The user sends the privilege set PF = $\{pj\}$ for the file F as well as the proof to the private cloud server. Upon receiving the request, the private cloud server first verifies the proof from the S-CSP. If it is passed, the private cloud server computes $\phi' F,pr$ = TagGen(ϕF, kpr)for all pr satisfying R(p, pr) = 1 foreach $p \in PF - PU$, which will be returned to the user. The user also uploads these tokens of the file F to the private cloud server. Then, the privilege set of the file is set to be the union of PF and the privilege sets defined by the other data owners.

3.File Retrieving

The user downloads his files in the same way as the deduplication system in Section 4.1. That is, the user canrecover the original file with the convergent key kF after receiving the encrypted data from the S-CSP.

V.MODULE DESCRIPTION

5.1Registration

The two secure channels are necessary for all two server, where a password is split into two parts, which are securely distributed to the two servers, respectively, during registration. Although we refer to the concept of public key cryptosystem,the encryption key of one server should be
unknown to another server.

Convergent Encryption Scheme

Each user has a private key x . Each user has three public keys:

Secure key size > 1024 bits (today even 2048 bits) prime modulus p, generator g
and public Y = g (mod p)

Novel is quite slow; it is used mainly for key authentication protocols

1.Deduplication

The tag of a file *F* will be determined by the file *F* and the privilege. To support authorized access, a secret key *kp*will be bounded with a privilege *p* to generate a file token.

Let $\phi'F,p$= TagGen(*F*, *kp*). The token $\phi'F,p$could only be computed by the users with privilege *p*. If a file has been uploaded by a user with a duplicate token $\phi'F,p$, then aduplicate check sent from another user will be successful if and only if he also has the file *F* and privilege *p*.

Token generation function could be easily implemented as *H*(*F*, *kp*).

2.Security on Attacks

Let k p be the user's private key. When an unauthorized user tries to crack kp it will check the authentication of the user. Hence it prevents the external attack

VI.IMPLEMENTATION

There are three basic entities in our project. A Client Program is the user who is used to carry out the file upload. A Private server program manages the private keys and the file token computations. A public server program is much similar to the S-CSP.

We are using the following functions.

- keygen.generate_Primekey(); - Generates the primary key for the registered user.
- elgamal.encrypt (Text,Common key, Public key, private key, privilege); - This function is used to encrypt the users' data with the given privileges. DupCheck (File) – This helps to deduplicate the files.

VII.EVALUATION

1.Deduplication Ratio

We first uploaded the first set of 50 files as an initial upload. For the second upload, we pick a portion of 50 files, from the initial set as duplicate files. As uploading and encryption would be skipped in case of duplicate files, the time spent on both of them decreases with increasing deduplication ratio. It was shown in (Fig. 2)

2.File size

To evaluate the effect of file size to the time spent on encryption of those files we upload 12 unique files, of particular file size and record the time break down. The time spent on encryption, upload increases linearly with the file size, since these operations involve the actual file data and incur file I/O with the whole file. It was shown in (Fig. 3)

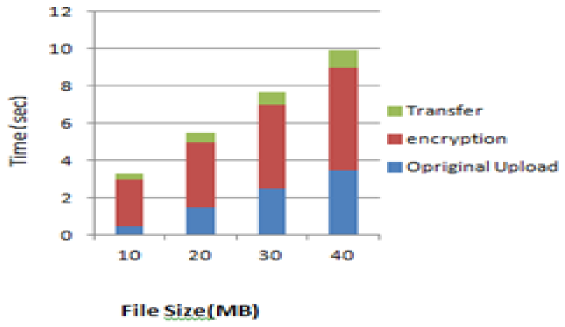

Fig 2.Different deduplication ratio

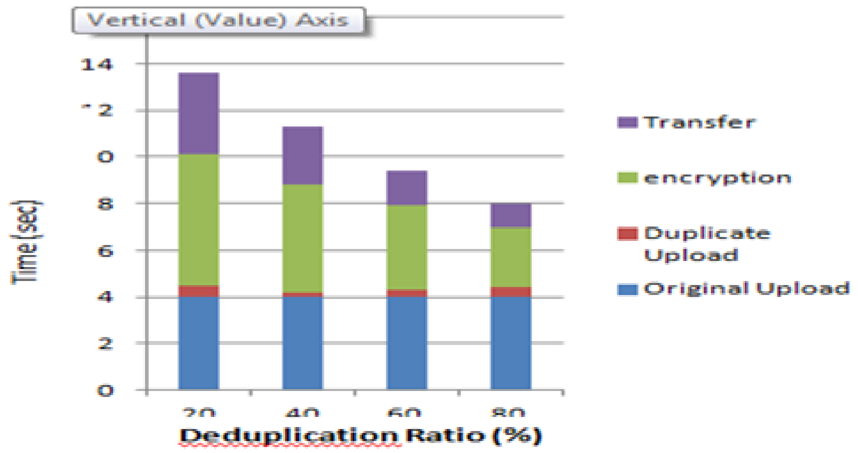

Fig 3.Time breakdown for different file size

VIII.CONCLUSION

In this paper, the notion of authorized data deduplication was proposed to protect the data by including differential privileges of users in the duplicate check. Here 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's and outsider's attacks specified in the proposed security model. As a proof of concept, we implemented a prototype of our proposed authorized duplicate check scheme. We showed that our authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and network transfer.

REFERENCE

P. Anderson and L. Zhang. Fast and secure laptop backups with encrypted de-duplication. In *Proc. of USENIX LISA*, 2010.M. Bellaire, S. Keelveedhi, and T. Ristenpart.Dupless: Serveraided encryption for deduplicated storage. In *USENIX Security Symposium*, 2013.

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

[2] R. D. Pietro and A. Sorniotti. Boosting efficiency and security in proof of ownership for deduplication. In H. Y. Youm and

[3] Y. Won, editors, *ACM Symposium on Information, Computer and Communications Security*, pages 81– 82. ACM, 2012.

[4] S. Quinlan and S. Dorward.Venti: a new approach to archival storage. In *Proc. USENIX FAST*, Jan 2002.

[5] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman- Peleg.Proofs of ownership in remote storage systems. In Y. Chen, G. Danezis, and V. Shmatikov, editors, *ACM Conference on Computer and Communications Security*, pages 491–500. ACM, 2011.

[6] J. Xu, E.-C.Chang, and J. Zhou. Weak leakage- resilient client-side deduplication of encrypted data in cloud storage. In *ASIACCS*, pages 195–206, 2013.

[7] M. W. Storer, K. Greenan, D. D. E. Long, and E. L. Miller.Securedatadeduplication.In *Proc. of StorageSS*, 2008.

[8] Z. Wilcox-O'Hearn and B. Warner. Tahoe: the least- authority filesystem. In *Proc. of ACM StorageSS*, 2008.

[9] J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl. A secure dat deduplication scheme for cloud storage. In *Technical Report*, 2013.