International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 02 Issue: 07 | Oct-2015 www.irjet.net p-ISSN: 2395-0072

DATA DE-DUPLICATION USING HYBRID CLOUD

Asst. Prof. S. B. Patil, Miss. Yashoda A. Kumbhar, Miss. Swapnali S. Mane

Asst. Prof, Department of Computer Science and Engineering, Shivaji University, Kolhapur, Maharashtra, India
Students, Department of Computer Science and Engineering, Shivaji University, Kolhapur, Maharashtra, India

---

Abstract - ***Cloud storage is one of the important technique for resources are sharing over the internet and in that to manage vast amount of data. A data de-duplication is one of the important technique for eliminating duplicate copies of repeating data[1]. We also present several new duplication construction for supporting authorized duplicate check in a hybrid clod architecture. Data de-duplication is one of important data compression techniques for eliminating duplicate copies of repeating data, and has been widely used in cloud storage to reduce the amount of storage space and save bandwidth. To protect the confidentiality of sensitive data while supporting de-duplication, the convergent encryption technique has been proposed to encrypt the data before outsourcing.***

Key Words: *- De-duplication, Hybrid Cloud, Encryption, Authorized Duplicate Check, etc…*

1. INTRODUCTION

Cloud storage provides a service for the evergreen management of vast amount of data in order to reduce the space and bandwidth [1]. De-duplication plays a vital role as a data or file compression technique which is most commonly used for eliminating repeated copies of data or file in cloud storage to reduce space and bandwidth de-duplication can takes place at either the file level or the block level.

For file level de-duplication it eliminates duplicate copies of the same file. While providing the data confidentiality the encryption requires different users to encrypt their data with their own keys.

Only the authorized user can access the files, upload the files and also download the files and other related data.

In today's world everyone uses the Internet most widely, so cloud computing is the most important technique, used by a Communication network like internet. So the business storage is done at very low cost. We can use cloud computing to store different types of data from many different fields like government, enterprise or anyone can store their personal data also. Cloud computing is nothing but sharing of resources so, without any background implementation details, users can share as well as access the different resources. As we know the most important issues of clod computing are related to memory management and the security of sensitive data. So the main drawback of cloud storage is data duplication which is increasing day by day. To reduce the memory management problem and to improve the storage space data de-duplication is an important technique that should be used by cloud computing. Recently all storage systems use the data de-duplication technique widely, so it is becoming most popular in many organizations.

Data compression also uses the same technique that is data de-duplication for reduce the data redundancy and going to store only single copy of that file. Data de-duplication is done by two ways one is File level and another is Block level. In File level approach we can eliminate the identical files from the storage space and in block level approach we can delete some amount of data i.e. the block of data from the files which are not similar Data de-duplication decrease the storage needs up to 90-95% backup application and for standard file system it is 68% But the main problem is security of data and privacy of that data form hackers. To secure the data from attackers users uses the encryption and decryption technique.

2. LITERATURE REVIEW

2.1 MD5 Algorithm

The MD5-message digest algorithm is a widely used cryptographic hash function producing a 128-bit(16byte)hash value, typically express in text format as a 32 digit hexadecimal number,MD5 has been utilized in wide variety of cryptographic application, and is also commonly used to verify data integrity.

The security of the MD5 hash function is severely compromised. A collision attack exists that can find collisions within seconds on a computer with a 2.6GHz Pentium 4 processor[2]. Further, there is also chosen-prefix collision attack that can produce a collision for two inputs with specified prefixes within hours, using off- the-shelf computing hardware.

2.2 Duplicate-check token

Assume a user with privilege p could forge a new duplicate-check token $\phi'_{F;p'}$ for any p' that does not match p. If it is a valid token, then it should be calculated as $\phi'_{F;p'} = H1(H(F), kp')$[2]. Recall that kp' is a secret key kept by the private cloud server and $H1(H(F), kp')$ is a valid message authentication code. Thus, without kp' , the adversary cannot forge and output a new valid one for any file *F[3]*.

For any user with privilege p, to output a new duplicate-check token $\phi'_{F;p}$, it also requires the knowledge of kp. Otherwise, the adversary could break the security of message authentication code

Formally, a convergent encryption scheme can be defined with **four primitive functions**:

• **KeyGenCE(*M*)** *: 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 cipher text *C*;

• **DecCE(*K,C*)** *: M* is the decryption algorithm that takes both the cipher text *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*).

3. PROPOSED WORK

To solve the problem of duplication we propose another advance de-duplication system supporting authorized duplicate check[4]. In this new system of de-duplication we use the concept of hybrid cloud. Hybrid cloud is combination of private cloud and public cloud.

The hybrid cloud structure is used to solve the problem. The private cloud stores the private keys of users and priviledge not issued by the user directly.

Tokens are generated for each file[3]. To get file token the user wants to send request to private cloud server. For performing the duplicate check user wants the token from private server.

Also the private cloud checks the user's identity before giving token to user. Authorized duplicate check performed by public cloud by the user before uploading his or her file. Based on result user can upload a file. If the user upload file having same name and same contents then one file can be deleted from the storage.

The authorized person or user can upload, send and download the files. The private cloud provides more security than public cloud. The encrypted file saved on public cloud.

4. DESIGN GOALS

Differential Authorization.

Each authorized user is able to get his/her individual token of his file to perform duplicate check 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.

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. 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[3]. The details are given below.

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 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.

Indistinguishability of file token/duplicate-check token.

It requires that any user without querying the private cloud server for some file token, he cannot get any useful information from the token, which includes the file information or the privilege information[9].

Data Confidentiality.

Unauthorized users without appropriate privileges or files including the S-CSP and the private cloud server,

should be prevented from access to the underlying plaintext stored at S-CSP[7]. In another word, the goal of the adversary is to retrieve and recover the files that do not belong to them. In our system, compared to the previous definition of data confidentiality based on convergent encryption, a higher level confidentiality is defined and achieved.

5. AUTHORIZED DE-DUPLICATION

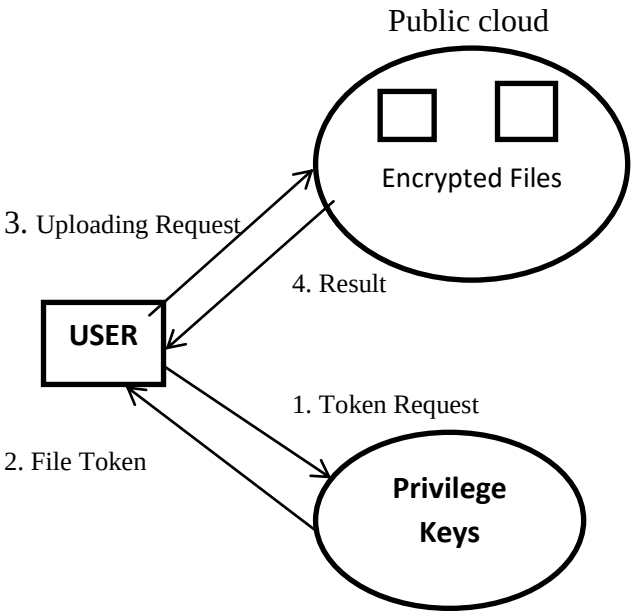

Fig 1: Architecture for Authorized De-duplication

User cannot access private key easily then kept and manage private cloud server. Privilege key accesses private cloud as user need purpose construction [2].

To perform, get a token file as user need send request to the private cloud server. Fig 1, Checks duplicate file, to get token file to the user need. Private cloud server also checks user identity before issuing corresponding token file to the user.

Authenticate duplicate check, can use file with public cloud server before uploading file. Fig 1. Result based on duplicate check, user either uploads this file.

1. User Module is the module in which the user are having authentication and security to access the detail which is offered in the ontology system before opening or examining the details users should have the account in that otherwise they should register first.
2. Secure De-duplication System to support authorized de-duplication, the tag of file „F‟ will be determined by the file „F‟ and the privilege .To show the difference with traditional notation tag, we call it file token. To support authorized access a secret key “kp” will be bounded with a privilege p to generate a file token. Let $‟F:p=TagGen(F,kp) denote the token of „F‟ that is only allowed to access by user with privilege „p‟[7]. In other words, the token $‟F:p could only be computed by the users with privilege „p‟. As a result if file is uploaded by a user with duplicate token $‟F:p then a duplicate check sent from another user will be successful if and only if he also has the file „F‟ and privilege „p‟. Such that a token generation function could be easily implemented as H(F,kp), where H(_) denotes the cryptographic function.
3. Security of Duplicate Check Token as we had consider several types of privacy so far and there is need to protect, that is, unforged ability of duplicate check token: There are two types of adversaries that is external adversary and internal adversary. As we know, the external adversary can be viewed as internal adversary without any privilege „p‟ on any file F where p doesn’t match p‟. Moreover it also requires that if the adversary does not make a request of token with its own privilege from private cloud server, it cannot counterfeit and output a valid duplicate token p on any F that has been queried.
4. Send Key once the key request is received, the sender can send the key or on the other hand the user 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.
We implement a prototype of the proposed authorized de-duplication system, in which we model three entities as separate C++ programs [8]. A Private Server program is used to model the private cloud which manages the private keys and handles the file token computation. A Client program is used to model the data users to carry out the file upload process. A Storage Server program is used to model the S-CSP which stores and de-duplicate files. Our implementation of the Client provides the following function calls to support token generation and de-duplication along the file upload process.

FileTag (File) – the process computes SHA-1 hash of the File as the File Tag;

TokenReq (Tag, UserID) – the process requests the Private Server for File Token generation with the User ID and the File Tag;

DupCheckReq (Token) – the process requests the Storage Server for the Duplicate Check of the File by sending the file token received from the private server;

ShareTokenReq (Tag, {Priv.}) – the process requests the Private Server to generate the Share File Token with Target Sharing Privilege Set and the File Tag;

FileEncrypt (File) – the process encrypts the File with Convergent Encryption using 256-bit of the AES algorithm

in cipher block chaining (CBC) mode, where the convergent key is from SHA-256 Hashing of the file;

FileUploadReq(FileID, File, Token) – the process uploads the File Data to the Storage Server if the file is Unique and the updates of the File Token is stored. Our implementation of the Private Server includes corresponding request handlers for the token generation and maintains a key storage with Hash Map.

TokenGen(Tag, UserID) – the process loads the associated privilege keys of the user and generate token with HMAC-SHA-1.

6. CONCLUSION

In this paper to study de-duplication technique with privilege access. The previous work and methodologies motivate us to develop our proposed system which will be the authorized data de-duplication technique [6]. We include differential privileges of users in the duplicate check to protect the data. 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.

7 . REFERANCES

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

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

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

[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.

[6] 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.

[7] 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.

[8] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer. Reclaiming space from duplicate files in a serverless distributed file system. In *ICDCS*, pages 617–624, 2002.

[9] D. Ferraiolo and R. Kuhn. Role-based access controls. In *15th NIST-NCSC National Computer Security Conf.*, 1992.

[10] 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.

[11] J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou. Secure deduplication with efficient and reliable convergent key management. In *IEEE Transactions on Parallel and Distributed Systems*, 2013.