ISSN: 2321-7782 (Online)
Volume 3, Issue 3, March 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Secure Authorized Deduplication Process in Hybrid Cloud

Mohamed Thoufeeq[1]
Department of Computer Science and Engineering
B.S. Abdur Rahman University
Vandalur, Chennai-48, India

Dr. Sharmila Sankar[2]
Professor and Head
Department of Computer Science and Engineering
B.S. Abdur Rahman University
Vandalur, Chennai-48, India

Dr. M. Sandhya[3]
Professor
Department of Computer Science and Engineering
B.S. Abdur Rahman University
Vandalur, Chennai-48, India

Abstract: ***Data deduplication is widely used in cloud services for data compression technique to eliminate the duplicate copies of the repeating data to minimize bandwidth and storage space. Convergent encryption technique is used to protect the confidentiality of the sensitive data for encrypting the data before outsourcing. Data deduplication eliminates extra copies by saving one data and replacing it with copies with pointers which leads back to the original data. Data deduplication is mostly used is industries for backup and disaster recovery applications. This paper uses convergent encryption technique to provide security to sensitive data using hybrid computing to authorize deduplication checks.***

Keywords: Deduplication, authorized duplicate check, confidentiality, hybrid cloud.

I. INTRODUCTION

Data deduplication is a specialized data compression technique for eliminating duplicate copies of same data. Deduplication techniques are used to improve storage utilization and can also be applied to network data transfers to reduce the number of bytes that must be sent. Unique chunks of data or byte patterns are identified during the process for deduplication and stored during a process of analysis. Three process of deduplication are Post-process deduplication, In-line deduplication and Source versus targeted deduplication. In general Cloud computing that deals with the network of remote servers on the internet to store, manage and process data, but rather than a local server or a personal computers.

One of the best examples for cloud storage is GMAIL. The management of increasing volume of data in cloud storage services is challengeable. To make data management scalable, Deduplication technique is used by the user to eliminate the duplicate copies of data. It includes convergent encryption, Identification protocol and proof of ownership.

Convergent encryption

Encryption and decryption of files are done by convergent encryption. From original copies convergent keys can be derived and that key is used for encryption. Duplicate data are checked from derived tag from the data copy. If the tags are the same it resembles each other and can be confirmed that both the files are same. The tag and convergent key are independently derived. Content hash keying (convergent key) produces identical cipher text from the identical plain text files respectively. To understand convergent key a simple implementation is given. Assuming Alice deriving her encryption key from her file F such that K = H (F), where H represents cryptographic hash function. This convergent encryption scheme is defined with four primitives,

- 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 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 ; and
- TagGen(M) -> T (M) is the tag generation algorithm that maps the original data copy M and outputs a tag T (M).

Proof of ownership (PoW)

PoW is a protocol which enables users to prove their ownership of the data copies to the storage server to avoid the duplicate copies. It is implemented by an interactive algorithm which is run by user and the storage server both acting as prover and verifier respectively. A short value is derived from the verifier ϕ (M) from the data copy M, to prove the ownership of the data copy M the prover needs to send the value ϕ to the verifier such that the value ϕ= ϕM. PoW as per content distribution network is that the attacker does not know the entire file but knows who have the file. PoW is specified by a summary function S (.) and an interactive two-party protocol || (P V). For solving the problem of small hash value as a proxy for the entire file, a solution is designed where the client proves the server that it has the original file. This proof mechanism that prevents such leakage is proof of ownership (PoW).

II. System Model

Architecture for secure authorized deduplication model is shown in Figure 1.Client deduplication scenario is where S-CSP (Storage Cloud Service Provider) server keeps a single copy of the original file even if any number of users requests to store the same file. All the users who have the original file use only the link of the single copy that is stored in the S-CSP server. This process is done when a client send the hash value of the original file to the server where the server checks whether the hash value is stores in the database. If the hash values are the same as the existing hash value that is stored, the server challenges the user for the proof of possession of the original file. Once the challenge is successful the client need not upload the file to the server once again but at the same time the server marks the client as the owner of the file. Now there won't be any difference between the user who uploaded the file and the client. This deduplication process saves the communication bandwidth and the storage space.

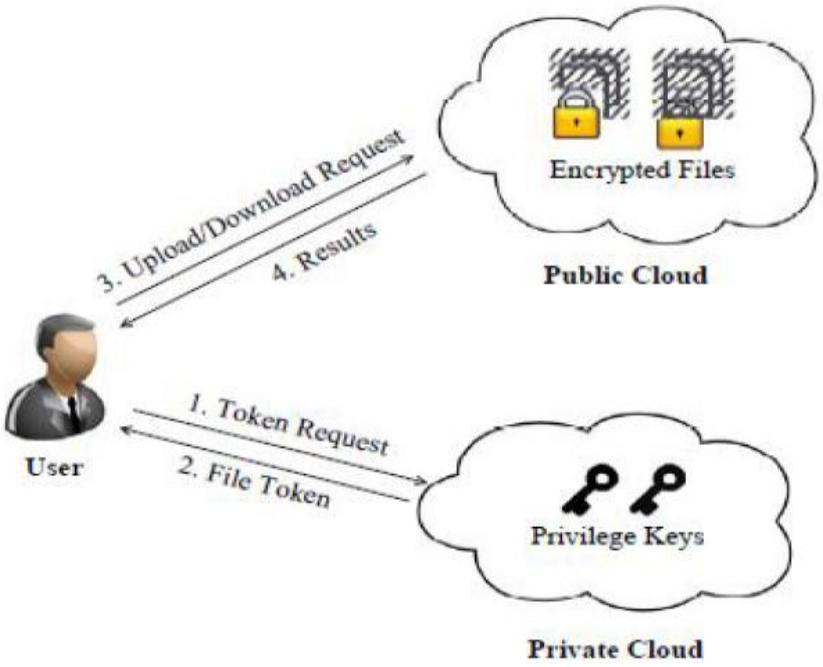

Figure 1. Architecture for Authorized Deduplication

Architecture for Secure Deduplication

- **S- CSP**: The main purpose of this entity is to work as a data storage service in the public cloud were half of the user S – CSP store the data. S- CSP on using data deduplication eliminates the duplicate data and keeps the unique data as the same and reduces the storage cost. S-CSP has more storage capacity and computational power. Users send representative token for accessing his file from public cloud and S-CSP matches the token internally and if matched alone it sends the file or encrypted cipher text with token. If this does not match then abort signal is sent to user. The decryption occurs after receiving the file using the convergent key KF.
- **Data user:** User is an entity who wants to access the data or file from S-CSP. The user generates the key and stores it in private cloud. In the storage system which supports duplication, the user uploads only unique data and not any duplicate data to save the bandwidth to upload, which is owned by the same or different user. All the files are protected by convergent key and can be accessed only by authorized persons. Here the user must register to private cloud for storing tokens with respective files which are stored in public cloud. Once the user wants to access the files from the private cloud, he access his representative token where the file consent is F and convergent key is KF respectively.
- **Private cloud:** To provide more security to the user private cloud is used in place of public cloud. The generated key of the users are stored in the private server. During the time of downloading the system checks for the key and then downloads it. The key cannot store internally to provide security. Private cloud only store the convergent key with respective file and when user wants to access the key he first, authority of user is checked and then provide key.
- **Public cloud:** Public cloud is mainly used for storage purpose. Public cloud is similar to S – CSP and user upload their files here. While the users wants to download a particular file, the server asks for the key that is generated are stored in the private cloud. Once the key matches that of the downloaded file, the download is initiated or else the user cannot access the file. Only authorized user have access the file. All the files in the public cloud are stored in encrypted file formats and even if unauthorized persons access the file he cannot decrypt it without the key that is generated or stored in the private cloud. In public cloud there are lots of files that are stored and each user access its respective file only if the token matches with S-CSP server token.

III. DESIGN GOALS

The problems of privacy preserving deduplication and a new deduplication system is proposed and the flaws are addressed.

- **Differential authorization –** Every authorized user can access their individual tokens of the file and perform duplication check of the file based on their authority. The above statement defines that a user cannot generate a token for duplication check for the file out of his own access or without the help of the private cloud server.
- **Authorized duplicate check –** Only authorized user are able to access their own token from the private cloud, but the public cloud performs duplication check and informs the user if there is a duplicate. The security of the file token and data file are very important. The two aspects of the file token security are Unforgeabilty and indistinguishability of the file token. The details of the security are given below
  - **Unforgeabilty of file token / duplicate – check token –** Registration in private cloud are made by the users for generating the file tokens. Users can use their respective token to upload or download the file on the public cloud. Users are blocked so that they cannot collude to public server to break the Unforgeabilty of the file token. S-CSP performs the duplication check honestly if the user request a duplication request. The duplication check must be issued from the private cloud of the user so that the security of the system is not compromised.

- **Indistinguishability of file token / duplicate –check token –** Users cannot get any useful information from the token unless he sends the query from his own private cloud. The user's information includes file and key information which can only be requested from a private cloud.

- **Data confidentiality – U**sers without appropriate tokens which include S-CSP and the private cloud server should be prevented from accessing the plaintext that is stored in the C-CSP. The goal of the adversary is to retrieve and recover the files that do not belong to them. Here a higher level of confidentiality is designed and achieved than the previous ways of making the system secure.

IV. PROPOSED STUDY

In the proposed system a hybrid cloud, a combination of both the private cloud and public cloud is implemented. Public cloud security cannot be processed as all the files can be access publicly which results in the loss of private data. Security of both private and public clouds systems are implemented to avoid deduplication of the files are increased. Duplicate copies of the files are avoided and users can only upload and download files from the public clouds. Only authorized users who have access to the private could can access the private cloud which makes the system secure and tokens are generated for each file to avoid deduplication which is specified before.

- **File uploading:** Flow chart for file uploading process is shown in Figure 2. When user upload the file to the public cloud the user first encrypt the file which is to be upload by the symmetric key and then send it to the public cloud and at the same time the user generates the key for the file and send it to the private cloud for uploading the file.

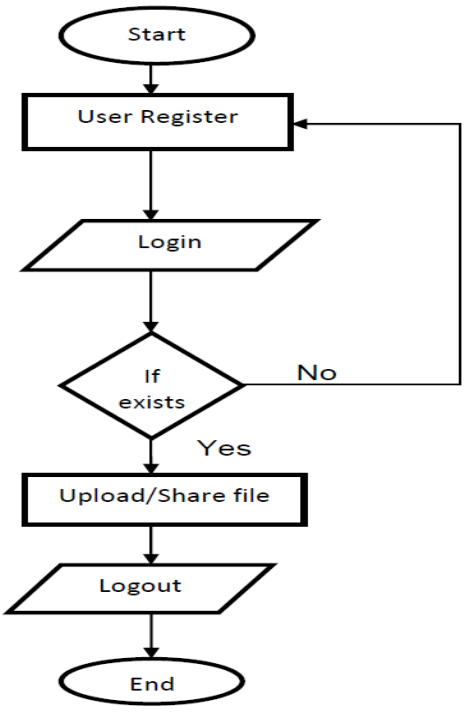

Figure 2. Flow Chart for File uploading

- **File downloading:** When user wants to download the file that is uploading on the public cloud.it make a request to the public cloud which provides a list of file that the users have in the cloud. Among these files the user selects the required file and request for the download. Now the private cloud sends a message with the key to the user through the private cloud so that it can download the file from the public cloud. If the key given by the private cloud is valid the required file can be downloaded or the user cannot download the file. When user wants to download the file from the public cloud it is in the encrypted format and then the user decrypts that file by using the same symmetric key. Flow chart for file downloading process is shown in Figure 3

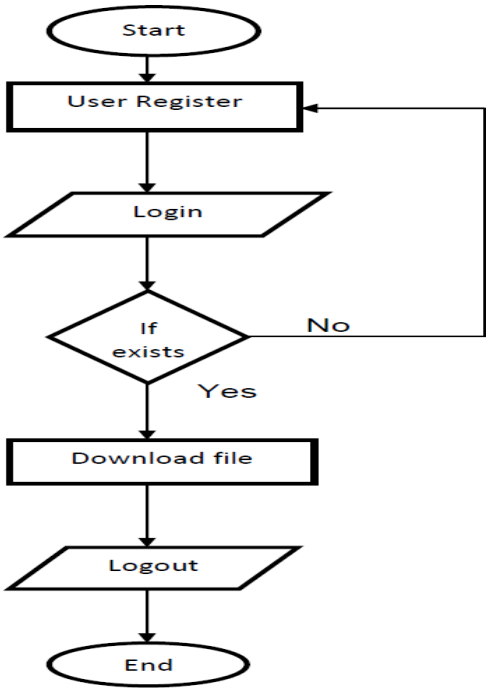

Figure 3. Flow Chart for File Downloading

V. IMPLEMENTATION

The implementation of the system depends upon three main entities. They are client program, private server program and the storage server program. The client server program manages the upload and download of the files by the user. The private cloud server program manages the private key and token computations. The storage server program maintains the S – CSP which stores and deduplicates files. The function calls that are used in the program are as follows.

- FileTag (File) - It computes SHA-1 hash of the File as File Tag.
- TokenReq (Tag, UserID) - It request the private server for File Token generation.
- DupCheckReq (Token) - It requests the Storage Server for Duplicate Check of the file.
- ShareTokenReq (Tag, {priv.}) - File token is generated for sharing the file token by requesting the private server.
- FileEncrypt (File) - It encrypts the File with Convergent Encryption using 256-bit AES algorithm in cipher block chaining (CBC) mode.
- FileUploadReq (FileID, File, Token) - File is uploaded to the Storage Server if the file is Unique and updates the File Token stored.
- TokenGen (Tag, UserID) - It loads the associated privilege keys of the user and generate the token with HMAC-SHA-1 algorithm.
- ShareTokenGen (Tag, {Priv.}) - Share token is generated with the corresponding privilege keys of sharing the privilege set with HMAC-SHA-1 algorithm.
- DupCheck (Token) - It searches the file to token map for duplicate.
- FileStore (FileID, File, Token) - It stores the File on Disk and updates the Mapping.

VI. RESULTS AND ANALYSIS

The test bed evaluation is conducted on the proposed model which usually focuses on comparing the overhead induced by authorization steps including file token generation and share token generation, against the convergent encryption and files

upload steps. The upload process involves Tagging, Token Generation, Duplicate Check, Share Token Generation, Encryption and Transfer of file is analyzed successfully by varying the factors such as File size, Number of stored files and Deduplication Ratio.

- Time spent for tagging, encryption which linearly increases the file size in uploading process.
- Token checking is done with the hash table and time taken for duplicate check remains stable for files stored in the server.
- The time spent for uploading and encryption decreases with increasing deduplication ratio. Then time spent for duplicate check also decreases when duplicate is found.

VII. CONCLUSION

To protect data security in cloud computing authorized data deduplication was proposed by including differential privileges to the users in the duplication check of the files. Several new deduplication constructions supporting authorized duplication check in hybrid cloud computing was performed. Security analysis demonstrates the schemes are secure in terms of both insider and outsider attacks specified in the proposed model. The prototype of the proposed model is given and experimental results are given to validate the security of the model. The results show that the authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and network transfer.

References

1. M. Bellare, S. Keelveedhi, and T. Ristenpart. Dupless, "Server-aided encryption for deduplicated storage", in proc. of USENIX Security Symposium, 2013.
2. J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou, "Secure deduplication with efficient and reliable convergent key management", in proc. of IEEE Transactions on Parallel and Distributed Systems, 2013.
3. Divyesh Minjrola, "A reverse deduplication storage system optimized for reads to latest backups", in proc. of APSYS, Apr 2013.
4. H. Xiong, A. Sorniotti, E. Androulaki, and L. Kencl, "A secure data deduplication scheme for cloud storage", in proc. Technical Report, 2013.
5. Y. Duan, Y. Wen, and H. Zhu, "Private data deduplication protocols in cloud storage", in proc. of the 27th Annual ACM Symposium on Applied Computing, pages 441–446. ACM, 2012.
6. A. Rahumed, H. C. H. Chen, Y. Tang, P. P. C. Lee, and J. C. S. Lui, "A secure cloud backup system with assured deletion and version control", in proc. 3rd International Workshop on Security in Cloud Computing, 2011.
7. S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems", In proc. ACM Conference on Computer and Communications Security, pages 491–500. ACM, 2011.
8. M. W. Storer, K. Greenan, D. D. E. Long, and E. L. Miller, "Secure data deduplication", in proc. of Storage SS, 2008.
9. Z. Wilcox-O'Hearn and B. Warner, "Tahoe: the least-authority file system", in proc. of ACM Storage SS, 2008.
10. J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer. Reclaiming space from duplicate files in a server less distributed file system. In ICDCS, pages 617–624, 2002.

