



A Review on Secure & Authorized Data De-Duplication in Hybrid Cloud

Trupti Deore, Prof. J. V. Shinde

Department of Computer Engineering, L.G.N Sapkal College, Nasik,
Maharashtra, India

Abstract— Now days as a result of advancement of storage technology and computer technology, larger fraction of information is being maintained in digitized kind. identical data is stored over and over again, consuming unnecessary storage space on the disc. Systems providing secured data storage are currently in larger demand. These systems give knowledge storage in a efficient manner. however a scenario could arise, once the data storage consists of huge quantity of duplicate and redundant data. These duplicate records could occupy extra space and access time. Hence, there's a necessity of banishing the duplicate records. Eliminating the duplicate records looks to be a simple task however needs lots of work to do because the duplicate records don't share any common key. Sometimes, errors occur as a results of transcription errors or incomplete data, lack of standard formats, or any combination of those errors. Deduplication is good for extremely redundant operations like backup, which needs repeatedly cope and storing the same data. Data deduplication is one amongst the foremost alive topics in storage as a result of it permits firms to avoid wasting lots of cash on storage prices. For cloud provider it's terribly useful as a result of you can deduplicate what you store. as a result of reduction in value it is being additional standard. This paper can in brief describe knowledge deduplication and provides a comprehensive survey.

Keywords— Deduplication, cloud storage, encryption, proof-of-ownership, revocation.

I. INTRODUCTION

Recently, many deduplication schemes are proposed to unravel this problem by permitting every owner to share a similar encryption key for a similar data. However, most of the schemes suffer from security flaws, since they are doing not contemplate the dynamic changes within the possession of outsourced data that occur often during a practical cloud storage service. Cloud computing permits access to resources from anyplace and at any time through the internet. the most advantage of using cloud storage from the customer's purpose of view is that customers will scale back their expenditure in buying and maintaining storage infrastructure whereas solely paying for the quantity of storage requested, which may be scaled-up and down upon demand. however it's also very true that cloud Storage isn't infinite. data deduplication is that the best way to handle these data. As customers ar involved concerning their personal data, they will encrypt their data before outsourcing so as to shield data privacy from unauthorized outside adversaries, still as from the cloud service supplier. this is often justified by current security trends and numerous industry rules like PCI DSS. However, conventional encryption makes deduplication not possible for the subsequent reason. Deduplication techniques benefit of data similarity to spot a similar data and scale back the storage space. In distinction, encryption algorithms disarrange the encrypted files so as to create ciphertext indistinguishable from in theory random data. Encryptions of a similar data by totally different users with different encryption keys leads to different ciphertexts, that makes it tough for the cloud server to see whether or not the plain data are a similar and deduplicate them.

II. LITERATURE SURVEY

Deduplication techniques are often classified into 2 different approaches: deduplication over unencrypted data and deduplication over encrypted data. In the former approach, most of the prevailing schemes have been proposed so as to perform a pow process in an efficient and sturdy manner, since the hash of the file, that is treated as a "proof" for the whole file, is susceptible to being leaked to outside adversaries because of its comparatively little size. Whereas, in the latter approach, data privacy is that the primary security requirement to guard against not solely outside adversaries however additionally within the cloud server. Thus, most of the schemes are proposed to provide data encryption, whereas still benefiting from a deduplication technique, by enabling data owners to share the encryption keys within the presence of the within and outside adversaries. Since encrypted data are given to a user, data access control are often in addition implemented by selective key distribution when the PoW process. However, not a lot of work has nonetheless been done to deal with dynamic possession management and its related security problem.

Harnick et al. [2] demonstrated how data deduplication technique will be used as a facet channel that reveals data to malicious users concerning the contents of files of alternative users. because the volume of data will increase, thus

will the demand for on-line storage services, from straightforward backup services to cloud storage infrastructures. Though deduplication is most effective once applied across multiple users, cross-user deduplication has serious privacy implications. Some straightforward mechanisms will modify cross-user deduplication whereas greatly reducing the danger of data leakage. Cloud storage refers to scalable and elastic storage capabilities delivered as a service using web technologies with elastic provisioning and usebased evaluation that does not penalise users for dynamic their storage consumption without notice.

On the idea of Harnick et al.'s study, Halevi et al. [3] conjointly introduced an identical attack situation on cloud storage that uses deduplication across multiple users. establish attacks that exploit client-side deduplication, permitting an attacker to achieve access to arbitrary-size files of alternative users based on a really small hash signatures of those files. additionally specifically, an attacker who is aware of the hash signature of a file will persuade the storage service that it owns that file, thus the server lets the attacker transfer the complete file. to beat such attacks, author introduce the notion of proofs-of-ownership (PoWs), that lets a consumer efficiently influence a server that that the client holds a file, instead of just some short data concerning it and formalize the idea of proof-of-ownership, below rigorous security definitions, and rigorous potency necessities of computer memory unit scale storage systems then present solutions supported Merkle trees and specific encodings, and analyze their security. we enforced one variant of the theme.

Authors in [4] explains that existing cryptographic techniques facilitate users make sure the privacy and integrity of files they retrieve. it's additionally natural, however, for users to need to verify that archives don't delete or modify files before retrieval. The goal of a POR is to accomplish these checks while not users having to download the files themselves. A POR may give quality-of-service guarantees, i.e., show that a file is recoverable at intervals an exact time bound.

Authors in [5]introduce a model for obvious information possession (PDP) that permits a client that has stored data at an untrusted server to verify that the server possesses the original information while not retrieving it. The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, that drastically reduces I/O prices. The client maintains a relentless quantity of metadata to verify the proof. The challenge/response protocol transmits alittle, constant quantity of data, that minimizes network communication. Thus, the PDP model for remote data checking supports massive data sets in widely-distributed storage system. However, proof of retrievability and data possession usually use a pre-processing step that can't be utilized in the data deduplication procedure. Despite their significant edges in terms of saving resources, these deduplication schemes could cause another security vulnerability and reveal users' personal data, above all, once partial info of users' data has already been leaked.

Authors in [6] present a mechanism to reclaim space from this incidental duplication to make it available for controlled file replication. Our mechanism includes: (1) convergent encryption, which enables duplicate files to be coalesced into the space of a single file, even if the files are encrypted with different users' keys; and (2) SALAD, a Self-Arranging Lossy Associative Database for aggregating file content and location information in a decentralized, scalable, fault-tolerant manner. Paper addresses the problems of identifying and coalescing identical files in the Farsite distributed filesystem, for the purpose of reclaiming storage space consumed by incidentally redundant content. Farsite is a secure, scalable, serverless file system that logically functions as a centralized file server but that is physically distributed among a networked collection of desktop workstation. They includes convergent encryption, which enables duplicate files to coalesced into the space of a single file, even if the files are encrypted with different users' keys, and SALAD, a Self- Arranging, Lossy, Associative Database for aggregating file content and location information in a decentralized, scalable, faulttolerant manner. It shows results from large-scale simulation experiments using file content data collected from a set of 585 desktop file systems.

we have a tendency to emphasize that "convergent encryption", that encrypts a file F using hash value $hash(F)$ as encryption key, isn't leakage-resilient and is so insecure within the setting of prisoner of war. Therefore, the direct combination of a pow scheme and convergent encryption isn't a solution for client-side deduplication over encrypted data.

Xu et al. [7] additionally introduced an analogous data integrity attack in the cloud storage service, known as a poison attack. In order to resolve this problem, Bellare et al. [8] introduced a message-locked encryption (MLE) concept and its security notion, and proposed irregular convergent encryption as one implementation of MLE. convergent encryption, are used to give data confidentiality in deduplication. A user uses original data copy to derive a convergent key and encrypt the data by using convergent encryption. User derives tag for every data copy. such to find duplicates tag are going to be used. If the 2 data copies are similar then tags are same. convergent encryption [6], provides data confidentiality in deduplication. A user (or data owner) derives a convergent key from every original data copy and encrypts the information copy with the convergent key. in addition, the user also derives a tag for the data copy, such the tag will be used to detect duplicates. Here, we have a tendency to assume that the tag correctness property [6] holds, i.e., if 2 data copies are identical, then their tags are identical. To find duplicates, the user first sends the tag to the server side to examine if the identical copy has been already stored. Note that both the convergent key and the tag are independently derived, and the tag cannot be used to deduce the convergent key and compromise data confidentiality. Both the encrypted data copy and its corresponding tag will be stored on the server side. Formally, a convergent encryption scheme can be defined with four primitive functions:

- $KeyGenCE(M) \rightarrow K$ is the key generation algorithm that maps a data copy M to a convergent key K ;
- $EncCE(K,M) \rightarrow C$ 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 ;
- $DecCE(K,C) \rightarrow M$ 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) \rightarrow T (M) is the tag generation algorithm that maps the original data copy M and outputs a tag T (M).

The user initial sends the tag to the server aspect to check if the identical copy has been already hold on for sight duplicates.[9]. convergent encryption is insecure within the setting of pow, wherever the hash price of the file (that is, a deterministic encryption key) could also be leaked [6],[2]. Unfortunately, this can be also the case in MLE [7] and Xu et al.'s schemes [6]. Since the hash price of the file is used as the KEK in each schemes, if the KEK is disclosed, adversaries who get it are able to decrypt the key encryption message and acquire the encryption key, although the encryption key is not deterministic. Another disadvantage in each schemes is the lack of dynamic ownership management among the data owners. for instance, suppose a group of users share data within the cloud storage. Some users might request data deletion or modification in the storage. Then, they should be prevented from accessing the original data after now instance (forward secrecy). Likewise, when a user subsequently uploads the data, access right to the previous data shouldn't be to him before that time instance (backward secrecy). However, in each schemes, this unauthorized knowledge access cannot be controlled, since the data encryption key cannot be updated in the least once its initial choice and distribution by an initial uploader.

Authors in [9] makes the primary attempt to formally address the problem of achieving efficient and reliable key management in secure deduplication. we have a tendency to initial introduce a baseline approach during which every user holds an independent master for encrypting the convergent keys and outsourcing them to the cloud. However, such a baseline key management scheme generates a huge variety of keys with the increasing variety of users and needs users to dedicatedly defend the master keys. to the present end, we have a tendency to propose Dekey , a new construction during which users don't have to be compelled to manage any keys on their own but instead firmly distribute the convergent key shares across multiple servers.

Shin et al [11]. proposed a deduplication scheme over encrypted data that uses predicate encryption. This approach allows deduplication only of files that belong to the same user, which severely reduces the effect of deduplication. Thus, in this paper, we focus on deduplication across different users such that identical files from different users are detected and deduplicated safely to provide more storage savings.

The proposed scheme ensures that only authorized access to the shared data is possible, which is considered to be the most important challenge for efficient and secure cloud storage services [12] in the environment where ownership changes dynamically.

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 partners who have the file. The partners follow the "bounded retrieval model", such that they can help the attacker obtain the file, subject to the constraint that they must send fewer bits than the initial min-entropy of the file to the attacker [13]

The traditional deduplication methods cannot be directly extended and applied in distributed and multi-server systems. If it is used, it cannot resist the collusion attack launched by multiple servers. In [14] a file is first split and encoded into fragments by using the technique of Ramp secret sharing, instead of encryption mechanisms. These shares are then distributed across multiple independent storage servers. Tag, a short cryptographic hash value of the content will also be computed and sent to each storage server as the fingerprint of the fragment stored at each server. Only the data owner who first uploads the data is required to compute and distribute such secret shares, while all following users who own the same data copy do not need to compute and store these shares any more. To recover data copies, users must access a minimum number of storage servers through authentication and obtain the secret shares to reconstruct the data

These dynamic ownership changes may occur very frequently in a practical cloud system, and thus, it should be properly managed in order to avoid the security degradation of the cloud service. However, the previous deduplication schemes could not achieve secure access control under a dynamic ownership changing environment, in spite of its importance to secure deduplication, because the encryption key is derived deterministically and rarely updated after the initial key derivation. Therefore, for as long as revoked users keep the encryption key, they can access the corresponding data in the cloud storage at any time, regardless of the validity of their ownership. This is the problem we attempt to solve in this study.

III. DISCUSSIONS

In cloud storage services, de-duplication technology is commonly used to reduce the space and bandwidth requirements of services by eliminating redundant data and storing only a single copy of them. De-duplication is most effective when multiple users outsource the same data to the cloud storage, but it raises issues relating to security and ownership. Proof-of-ownership schemes allow any owner of the same data to prove to the cloud storage server that he owns the data in a robust way. However, many users are likely to encrypt their data before outsourcing them to the cloud storage to preserve privacy, but this hampers de-duplication because of the randomization property of encryption.

The solution allows the cloud storage server to obtain the outsourced plain data, which may violate the privacy of the data if the cloud server cannot be fully trusted. Convergent encryption resolves this problem effectively However, convergent encryption suffers from security flaws with regard to tag consistency and ownership revocation. Because of the actual data storage directly stored in the cloud server hence more data has been loss. But to overcome this problem in the proposed system stored secrets on cloud server ,so no data has been loss

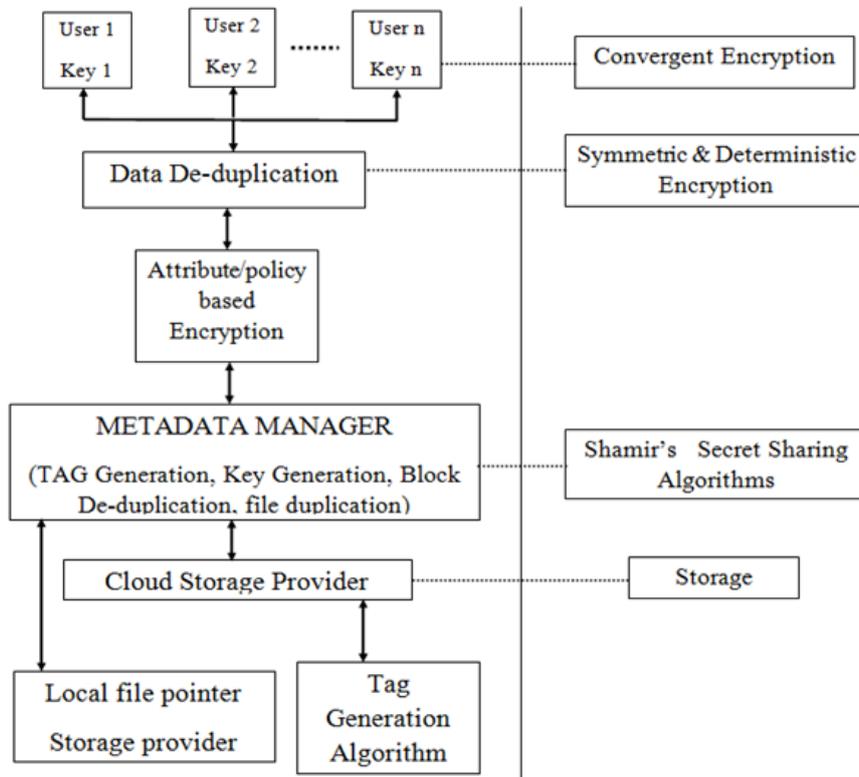


Fig 1. Proposed Architecture

IV. CONCLUSIONS

Managing encrypted data with de-duplication is important and significant in practice for achieving a successful cloud storage services, especially for big data storage. In this paper, we proposed a practical scheme to manage the encrypted data in cloud with de-duplication with based on ownership challenge. Cloud storage system for organizations to outsource backup and archival storage to public cloud vendors, with three goals in mind: reliability, security, cost efficiency. We also address the tag ownership and data ownership management in cloud storage.

ACKNOWLEDGMENT

Our sincere thanks go to KCTs Late G.N. Sapkal College of Engineering for providing a strong platform to develop our skill and capabilities. I would like to thanks all those who directly or indirectly help us in presenting the paper. I hereby take this opportunity to express our heartfelt gratitude towards the people whose help is very useful to complete our project. I would like to express our heartfelt thanks to my HOD Prof N.R.Wankhade and guide Prof. J.V.Shinde whose experienced guidance became very valuable for us.

REFERENCES

- [1] Hur, Junbeom, et al. *Secure data deduplication with dynamic ownership management in cloud storage*. IEEE Transactions on Knowledge and Data Engineering 28.11 (2016): 3113-3125.
- [2] Harnik, Danny, Benny Pinkas, and Alexandra Shulman-Peleg. *Side channels in cloud services: Deduplication in cloud storage*. IEEE Security & Privacy 8.6 (2010): 40-47.
- [3] Halevi, Shai, et al. *Proofs of ownership in remote storage systems*. Proceedings of the 18th ACM conference on Computer and communications security. ACM, 2011.
- [4] Juels, Ari, and Burton S. Kaliski Jr. *PORs: Proofs of retrievability for large files*. Proceedings of the 14th ACM conference on Computer and communications security. Acm, 2007.
- [5] Ateniese, Giuseppe, et al. *Provable data possession at untrusted stores*. Proceedings of the 14th ACM conference on Computer and communications security. Acm, 2007.
- [6] Douceur, John R., et al. *Reclaiming space from duplicate files in a serverless distributed file system*. Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on. IEEE, 2002.
- [7] Xu, Jia, Ee-Chien Chang, and Jianying Zhou. *Leakage-resilient client-side deduplication of encrypted data in cloud storage*. IACR ePrint Archive, 15pages (2011).
- [8] M. Bellare, S. Keelveedhi, and T. Ristenpart. *Message-locked encryption and secure deduplication*. In *EUROCRYPT*, pages 296– 312, 2013.
- [9] 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.

- [10] Li, Jin, et al. *Secure deduplication with efficient and reliable convergent key management*. IEEE transactions on parallel and distributed systems 25.6 (2014): 1615-1625.
- [11] Y. Shin and K. Kim, *Equality predicate encryption for secure data deduplication*, Proc. Conference on Information Security and Cryptology (CISC-W), pp. 64–70, 2012.
- [12] M. Mulazzani, S. Schrittwieser, M. Leithner, and M. Huber, *Dark clouds on the horizon: using cloud storage as attack vector and online slack space*, Proc. USENIX Conference on Security, 2011
- [13] S. Halevi, D. Harnik, B. Pinkas, and A. ShulmanPeleg. *Proofs of ownership in remote storage systems*. In Y. Chen, G. Danezis, and V. Shmatikov, editors, ACMConference on Computer and CommunicationsSecurity, pages 491–500. ACM, 2011
- [14] Jin Li, Xiaofeng Chen, Xinyi Huang, Shaohua Tang and Yang Xiang, *Secure Distributed System with Improved Reliability* in IEEE Transaction on Computers Volume: PP Year 2015.