Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, May : 2024

# ENSURING DATA INTEGRITY AND ELIMINATING REDUNDANCY IN CLOUD STORAGE

**Mr.Ch.Aravind Kumar**, Assistant professor CSE, Vaagdevi College of Engineering(Autonomous),India
**Guggilla Priyanka** ,UG Student ,CSE, Vaagdevi College of Engineering(Autonomous),India
**Orsu Ruben** ,UG Student ,CSE, Vaagdevi College of Engineering(Autonomous),India
**Maskula Sahith** ,UG Student ,CSE, Vaagdevi College of Engineering(Autonomous),India

## ABSTRACT

As the cloud computing technology develops during the last decade, outsourcing data to cloud service for storage becomes an attractive trend, which benefits in sparing efforts on heavy data maintenance and management. Nevertheless, since the outsourced cloud storage is not fully trustworthy, it raises security concerns on how to realize data deduplication in cloud while achieving integrity auditing. In this work, we study the problem of integrity auditing and secure deduplication on cloud data. Specifically, aiming at achieving both data integrity and deduplication in cloud, we propose two secure systems, namely SecCloud and SecCloud+. SecCloud introduces an auditing entity with a maintenance of a MapReduce cloud, which helps clients generate data tags before uploading as well as audit the integrity of data having been stored in cloud. Compared with previous work, the computation by user in SecCloud is greatly reduced during the file uploading and auditing phases. SecCloud+ is designed motivated by the fact that customers always want to encrypt their data before uploading, and enables integrity auditing and secure deduplication on encrypted data.

## 1. INTRODUCTION

Cloud storage is a model of networked enterprise storage where data is stored in virtualized pools of storage which are generally hosted by third parties. Cloud storage provides customers with benefits, Ranging from cost saving and simplified convenience, to mobility opportunities and scalable service. These great features attract more and more customers to utilize and storage their personal data to the cloud storage: according to the analysis report, the volume of data in cloud is expected to achieve 40 trillion gigabytes in 2020. Even though cloud storage system has been widely adopted, it fails to accommodate some important emerging needs such as the abilities of auditing integrity of cloud files by cloud clients and detecting duplicated files by cloud servers. We illustrate both problems below. The first problem is integrity auditing. The cloud server is able to relieve clients from the heavy burden of storage management and maintenance. The most difference of cloud storage from traditional in-house storage is that the data is transferred via Internet and stored in an uncertain domain, not under control of the clients at all, which inevitably raises [1] clients great concerns on the integrity of their data. These concerns originate from the fact that the cloud storage is susceptible to security threats from both outside and inside of the cloud , and the uncontrolled cloud servers may passively hide some data loss incidents from the clients to maintain their reputation. The second problem is secure de-duplication. The rapid adoption of cloud services is accompanied by increasing volumes of data stored at remote cloud servers. Among these remote stored files, most of them are duplicated: according to a recent survey by EMC, 75% of recent digital data is duplicated copies. This fact raises a technology namely deduplication, in which the cloud servers would like to de-duplicate by keeping only a single copy for each file and make a link to the file for every client who owns or asks to store the same file. Unfortunately, this action of deduplication would lead to a number of threats potentially affecting the storage system, for example, a server telling a client that it does not need to send the file reveals that some other client has the exact same file, which could be sensitive sometimes. These attacks originate from the reason that the proof that the client owns a given file is solely based on a static, short value. Thus, the second problem is generalized as how can the cloud servers efficiently confirm that the client owns the uploaded file before creating a link to this file for him/her.

**2. Literature Survey**

Data integrity and storage efficiency are two important requirements for cloud storage. Proof of Retrievability (POR) and Proof of Data Possession (PDP) techniques assure data integrity for cloud storage. Proof of Ownership (POW) [2] improves storage efficiency by securely removing unnecessarily duplicated data on the storage server. However, trivial combination of the two techniques, in order to achieve both data integrity and storage efficiency, results in non-trivial duplication of metadata (i.e., authentication tags), which contradicts the objectives of POW. Recent attempts to this problem introduce tremendous computational and communication costs and have also been proven not secure. It calls for a new solution to support efficient and secure data integrity auditing with storage deduplication for cloud storage. In this paper we solve this open problem with a novel scheme based on techniques including polynomial-based authentication tags and homomorphic linear authenticators. Our design allows deduplication of both files and their corresponding authentication tags. Data integrity auditing and storage deduplication are achieved simultaneously. Our proposed scheme is also characterized by constant realtime communication and computational cost on the user side. Public auditing and batch auditing are both supported. Hence, our proposed scheme outperforms existing POR and PDP schemes while providing the additional functionality of deduplication. We prove the security of our proposed scheme based on the Computational Diffie-Hellman problem, the Static Diffie-Hellman problem and the t-Strong Diffie-Hellman problem. Numerical analysis and experimental results on Amazon AWS show that our scheme is efficient and scalable.

Data integrity and storage efficiency are two important requirements for cloud storage. Proof of Retrievability (POR) and Proof of Data Possession (PDP) techniques assure data integrity for cloud storage. Proof of Ownership (POW) [3] improves storage efficiency by securely removing unnecessarily duplicated data on the storage server. However, trivial combination of the two techniques, in order to achieve both data integrity and storage efficiency, results in non-trivial duplication of metadata (i.e., authentication tags), which contradicts the objectives of POW. Recent attempts to this problem introduce tremendous computational and communication costs and have also been proven not secure. It calls for a new solution to support efficient and secure data integrity auditing with storage deduplication for cloud storage. In this paper we solve this open problem with a novel scheme based on techniques including polynomial-based authentication tags and homomorphic linear authenticators. Our design allows deduplication of both files and their corresponding authentication tags. Data integrity auditing and storage deduplication are achieved simultaneously. Our proposed scheme is also characterized by constant realtime communication and computational cost on the user side. Public auditing and batch auditing are both supported. Hence, our proposed scheme outperforms existing POR and PDP schemes while providing the additional functionality of deduplication. We prove the security of our proposed scheme based on the Computational Diffie-Hellman problem, the Static Diffie-Hellman problem and the t-Strong Diffie-Hellman problem. Numerical analysis and experimental results on Amazon AWS show that our scheme is efficient and scalable.

Cloud storage systems are becoming increasingly popular. A promising technology that keeps their cost down is deduplication, which stores only a single copy of repeating data. Client-side deduplication attempts to identify deduplication opportunities already at the client and save the bandwidth of uploading copies of existing files to the server. In this work we identify attacks that exploit client-side deduplication, allowing an attacker to gain access to arbitrary-size files of other users based on a very small hash signatures of these files. More specifically, an attacker who knows the hash signature of a file can convince the storage service that it owns that file, hence the server lets the attacker download the entire file. (In parallel to our work, a subset of these attacks were recently introduced in the wild with respect to the Dropbox file synchronization service.) To overcome such attacks, we introduce the notion of proofs-ofownership (PoWs), which lets a client efficiently prove to a server that that the client holds a file, rather than just some short information about it. We formalize the concept of proof-of-ownership, under rigorous security definitions, and rigorous efficiency requirements of Petabyte scale storage systems. We then present solutions based on Merkle trees and specific encodings, and analyze their security. We implemented one variant of the scheme. Our performance measurements indicate that the scheme incurs only a small overhead compared to naive client-side deduplication.

Cloud storage service providers such as Dropbox, Mozy, and others perform deduplication to save space by only storing one copy of each file uploaded. Should clients conventionally encrypt their files, however, savings are lost. Message-locked encryption (the most prominent manifestation of which is convergent encryption) resolves this tension. However it is inherently subject to brute-force attacks that can recover files falling into a known set. We propose an architecture that provides secure deduplicated storage resisting brute-force attacks, and realize it in a system called DupLESS. In DupLESS, clients encrypt under message-based keys obtained from a key-server via an oblivious PRF protocol. It enables clients to store encrypted data with an existing service, have the service perform deduplication on their behalf, and yet achieves strong confidentiality guarantees. We show that encryption for deduplicated storage can achieve performance and space savings close to that of using the storage service with plaintext data.

We introduce a model for *provable data possession* (PDP) that allows a client that has stored data at an untrusted server to verify that the server possesses the original data without retrieving it. The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces I/O costs[4]. The client maintains a constant amount of metadata to verify the proof. The challenge/response protocol transmits a small, constant amount of data, which minimizes network communication. Thus, the PDP model for remote data checking supports large data sets in widely-distributed storage system.
We present two provably-secure PDP schemes that are more efficient than previous solutions, even when compared with schemes that achieve weaker guarantees. In particular, the overhead at the server is low (or even constant), as opposed to linear in the size of the data. Experiments using our implementation verify the practicality of PDP and reveal that the performance of PDP is bounded by disk I/O and not by cryptographic computation.

## 3. PROBLEM STATEMENT

- Ateniese et al. proposed a dynamic PDP schema but without insertion operation.
- Erway et al. improved Ateniese et al.'s work and supported insertion by introducing authenticated flip table.
- Wang et al.proposed proxy PDP in public clouds.
- Zhu et al. proposed the cooperative PDP in multi-cloud storage.
- Wang et al. improved the POR model by manipulating the classic Merkle hash tree construction for block tag authentication.
- Xu and Chang proposed to improve the POR schema with polynomial commitment for reducing communication cost.
- Stefanov et al. proposed a POR protocol over authenticated file system subject to frequent changes.
- Azraoui et al. combined the privacy-preserving word search algorithm with the insertion in data segments of randomly generated short bit sequences, and developed a new POR protocol.
- Li et al. considered a new cloud storage architecture with two independent cloud servers for integrity auditing to reduce the computation load at client side.

**LIMITATIONS**

- The first problem is integrity auditing. The cloud server is able to relieve clients from the heavy burden of storage management and maintenance. The most difference of cloud storage from traditional in-house storage is that the data is transferred via Internet and stored in an uncertain domain, not under control of the clients at all, which inevitably raises clients great concerns on the integrity of their data.

- The second problem is secure deduplication. The rapid [5] adoption of cloud services is accompanied by increasing volumes of data stored at remote cloud servers. Among these remote stored files, most of them are duplicated: according to a recent survey by EMC, 75% of recent digital data is duplicated copies.
- Unfortunately, this action of deduplication would lead to a number of threats potentially affecting the storage system, for example, a server telling a client that it (i.e., the client) does not need to send the file reveals that some other client has the exact same file, which could be sensitive sometimes. These attacks originate from the reason that the proof that the client owns a given file (or block of data) is solely based on a static, short value (in most cases the hash of the file).

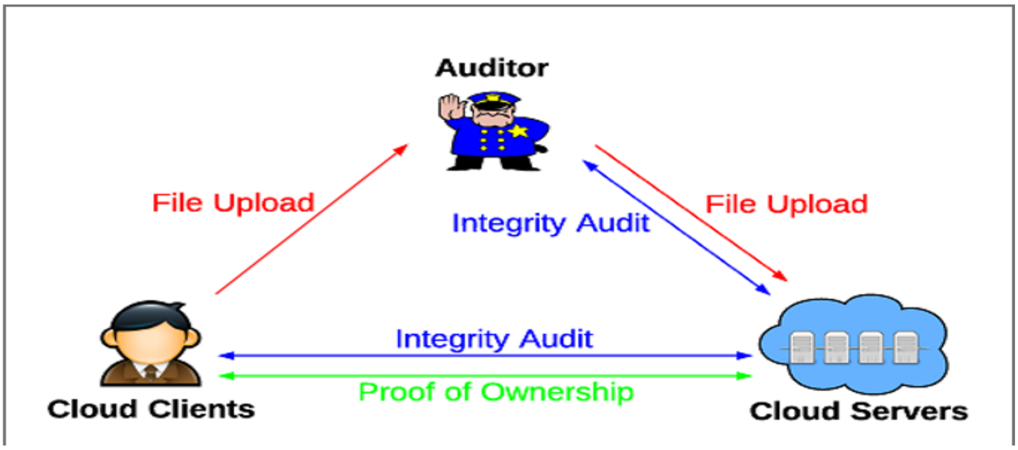
## 4. PROPOSED SYSTEM

- In this paper, aiming at achieving data integrity and deduplication in cloud, we propose two secure systems namely SecCloud and SecCloud+.
- SecCloud introduces an auditing entity with maintenance of a MapReduce cloud, which helps clients generate data tags before uploading as well as audit the integrity of data having been stored in cloud.
- Besides supporting integrity auditing and secure deduplication, SecCloud+ enables the guarantee of file confidentiality.
- We propose a method of directly auditing integrity on encrypted data.

## ADVANTAGES

- This design fixes the issue of previous work that the computational load at user or auditor is too huge for tag generation. For completeness of fine-grained, the functionality of auditing designed in SecCoud is supported on both block level and sector level. In addition, SecCoud also enables secure deduplication.
- The challenge of deduplication on encrypted is the prevention of dictionary attack.
- Our proposed SecCloud system has achieved both integrity auditing and file deduplication.

## 5. SYSTEM ARCHITECTURE




## 6. IMPLEMENTATION

### Cloud Service Provide

- In this module, we develop Cloud Service Provider module. This is an entity that provides a data storage service in public cloud.
- The CS provides the data outsourcing service and stores data on behalf of the users.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, May : 2024

- To reduce the storage cost, the CS eliminates the storage of redundant data via deduplication and keeps only unique data.
- In this paper, we assume that CS is always online and has abundant storage capacity and computation power.

**Data Users Module**

- 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 uploads unique data but does not upload any duplicate data to save the upload bandwidth, which may be owned by the same user or different users.
- In the authorized deduplication system, each user is issued a set of privileges in the setup of the system. Each file is protected with the convergent encryption key and privilege keys to realize the authorized deduplication with differential privileges.

**Auditor**

Auditor which helps clients upload and audit their outsourced data maintains a MapReduce cloud and acts like a certificate authority. This assumption presumes that the auditor is associated with a pair of public and private keys. Its public key is made available to the other entities in the system. The first design goal of this work is to provide the capability of verifying correctness of the remotely stored data. public verification, which allows anyone, not just the clients originally stored the file, to perform verification.

**Secure De-duplication System**

- We consider several types of privacy we need protect, that is, i) unforgeability of duplicate-check token: There are two types of adversaries, that is, external adversary and internal adversary.
- As shown below, the external adversary can be viewed as an internal adversary without any privilege.
- If a user has privilege $p$, it requires that the adversary cannot forge and output a valid duplicate token with any other privilege $p'$ on any file $F$, where $p$ does not match $p'$. Furthermore, it also requires that if the adversary does not make a request of token with its own privilege from private cloud server, it cannot forge and output a valid duplicate token with $p$ on any $F$ that has been queried.

## 7. EXPECTED OUTCOMES


**Fig-7.1 Result**


**Fig-7.2 Auditor Login**


**Fig-7.3 Cloud Server login**


**Fig-7.3.1 Cloud Server login**

**Fig-7.4 Cloud user login**

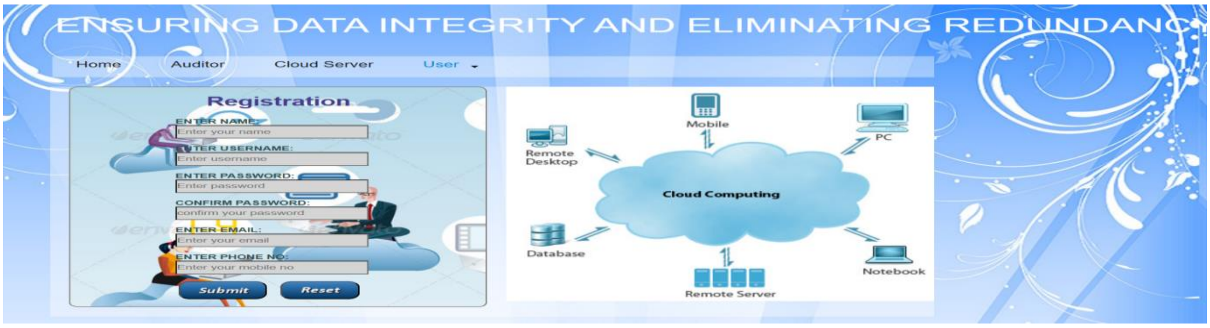



**Fig-7.5 Registration**

## 8.CONCLUSION

Aiming at achieving both data integrity and deduplication in cloud, we propose SecCloud and SecCloud+. SecCloud introduces an auditing entity with maintenance of a MapReduce cloud, which helps clients generate data tags before uploading as well as audit the integrity of data having been stored in cloud. In addition, SecCoud enables secure deduplication through introducing a Proof of Ownership protocol and preventing the leakage of side channel information in data deduplication. Compared with previous work, the computation by user in SecCloud is greatly reduced during the file uploading and auditing phases. SecCloud+ is an advanced construction motivated by the fact that customers always want to encrypt their data before uploading, and allows for integrity auditing and secure deduplication directly on encrypted data.

### 8.1 Future Scope

In the future, this project could expand its capabilities by integrating advanced security measures such as encryption and access controls to further safeguard data. Additionally, leveraging AI and machine learning algorithms could enable real-time analysis of data patterns and detection of anomalies, enhancing the system's ability to maintain integrity. Automation of redundancy management processes could streamline operations, while advanced disaster recovery solutions like geo-replication and failover mechanisms could ensure high availability and resilience. Scalability and performance optimization efforts would focus on handling increasing data volumes efficiently.

## 9. REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “A view of cloud computing,” *Communication of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[2] J. Yuan and S. Yu, “Secure and constant cost public cloud storage auditing with deduplication,” in *IEEE Conference on Communications and Network Security (CNS)*, 2013, pp. 145–153.

[3] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, “Proofs of ownership in remote storage systems,” in *Proceedings of the 18th ACM Conference on Computer and Communications Security*. ACM, 2011, pp. 491–500.

[4] S. Keelveedhi, M. Bellare, and T. Ristenpart, “Dupless: Serveraided encryption for deduplicated storage,” in *Proceedings of the 22Nd USENIX Conference on Security*, ser. SEC’13. Washington, D.C.: USENIX Association, 2013, pp. 179–194. [Online]. Available: https://www.usenix.org/conference/usenixsecurity13/technicalsessions/presentation/bellare

[5] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable data possession at untrusted stores,” in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, ser. CCS ’07. New York, NY, USA: ACM, 2007, pp. 598–609.