International Journal of Engineering Research and Science & Technology

IJERST

ISSN:2319-5991

www.ijerst.org

E-mail: editor@ijerst.org or ijerst.editor@gmail.com

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

ENSURING DATA INTEGRITY AND ELIMINATING REDUNDANCY IN CLOUD STORAGE

1Mrs. M. SUDA RANI, 2K. AKHILESH, 3S. KOUSHIK

4CP. HARISH , 5G. MEGHANA

1Assistant Professor, Department of CSIT, Sri Indu College of Engineering and Technology-Hyderabad

2345Under Graduate, Department of CSIT, Sri Indu College of Engineering and Technology-Hyderabad

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.

I. INTRODUCTION

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the common use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation. Cloud computing consists of hardware and software resources made available on the Internet as managed third-party services. These services typically provide access to advanced software applications and high-end networks of server computers.

The goal of cloud computing is to apply traditional supercomputing, or high- performance computing power, normally used by military and research facilities, to perform tens of trillions of computations per second, in consumer-oriented applications such as financial portfolios, to deliver personalized information, to provide data storage or to power large, immersive computer games.

The cloud computing uses networks of large groups of servers typically running low- cost consumer PC technology with specialized connections to spread data-processing chores across them. This shared IT infrastructure contains large pools of systems that are linked together. Often, virtualization techniques are used to maximize the power of cloud computing.

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

Cloud Computing comprises three different service models, namely Infrastructure- as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). The three service models or layer are completed by an end user layer that encapsulates the end user perspective on cloud services. The model is shown in figure below. If a cloud user accesses services on the infrastructure layer, for instance, she can run her own applications on the resources of a cloud infrastructure and remain responsible for the support, maintenance, and security of these applications herself. If she accesses a service on the application layer, these tasks are normally taken care of by the cloud service provider.

- On-demand self-service: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service's provider.
- Broad network access: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).

Resource pooling: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.

- Rapid elasticity: Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.

II. LITERATURE SURVEY

TITLE: A view of cloud computing

AUTHORS: M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia

ABSTRACT: Cloud computing, the long-held dream of computing as a utility, has the potential to transform a large part of the IT industry, making software even more attractive as a service and shaping the way IT hardware is designed and purchased. Developers with innovative ideas for new Internet services no longer require the large capital outlays in hardware to deploy their service or the human expense to operate it. They need not be concerned about overprovisioning for a service whose popularity does not meet their predictions, thus wasting costly resources, or underprovisioning for one that becomes wildly popular, thus missing potential customers and revenue. Moreover, companies with large batch-oriented tasks can get results as quickly as their programs can scale, since using 1,000 servers for one hour costs no more than using one server for 1,000 hours. This elasticity of resources, without paying a premium for large scale, is unprecedented in the history of IT.

TITLE: Secure and constant cost public cloud storage auditing with deduplication

AUTHORS: J. Yuan and S. Yu

ABSTRACT: 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) 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,

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

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.

TITLE: Proofs of ownership in remote storage systems

AUTHORS: S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg

ABSTRACT: 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.

TITLE: - Provable data possession at untrusted stores. AUTHORS: - G. Ateniese et al.

ABSTRACT: - 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. In particular, the overhead at the server is low (or even constant), as opposed to linear in the size of the data.

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

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.

III. SYSTEM ANALSIS

EXISTING SYSTEM

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.

DISADVANTAGES

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 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).

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

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

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.

IV. IMPLEMENTATION AND RESULTS

Cloud Service Provider

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.

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

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.

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

V. 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.

FUTURE SCOPE

To ensure data integrity and eliminate redundancy in cloud storage, various strategies can be employed, including data redundancy techniques, data integrity measures, and emerging trends like AI and cloud repatriation. Best practices involve identifying critical components, assessing redundancy requirements, and regular testing and maintenance, with further research needed to explore emerging technologies like blockchain and advanced AI-powered data management systems.

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

ISSN 2319-5991 www.ijerst.com

Vol. 21, Issue 2, 2025

22Nd USENIX Conference on Security, ser. SEC'13. Washington, D.C.: USENIX Association, 2013, pp. 179–194.

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.

6. G. Ateniese, R. Burns, R. Curtmola, J. Herring, O. Khan, L. Kissner, Z. Peterson, and D. Song, "Remote data checking using provable data possession," ACM Trans. Inf. Syst. Secur., vol. 14, no. 1, pp. 12:1–12:34, 2011.

7. G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks, ser. SecureComm '08. New York, NY, USA: ACM, 2008, pp. 9:1–9:10.