Identifying Suspicious Cloud File Migration or Replication

[1]Kapudasi Sindhuja, [2]S Srinivas
[1,2]Dept. of Computer Science & Engineering, KIET, Kakinada, AP, India

Abstract
Now-a-days all are using the cloud server to store their data and provides many features suitable for the users or customers. We have cloud servers like Google Cloud Platform, Microsoft Azure etc.For cloud also sometimes there will be storage problem to store the data of the users.We need the security to the data stored in the cloud.For hospitals and some private companies the data should be secure and confidential.So we need both storage and the security to our data stored in the cloud.
Client-Side Authorized Deduplication Here, it is suggested to use CP-ABE, which offers cloud-based security and deduplication. The proposed system offers data security in an encrypted format. In this system, we encrypted user data uploaded to the cloud with the CP-ABE algorithm using the user's attributes. Additionally, it examines any file duplication on the cloud. When a file is deduplicated, the server forbids uploading an already-existing copy of the file. Deduplication assists in releasing cloud storage. The suggested scheme has benefits over current schemes and satisfies the security criteria. In the cloud context, the suggested permitted deduplication technique offers a good trade-off between storage space efficiency and security, and it is ideal for the hybrid cloud architecture.

I. Introduction

CSP can get rid of seldom-used information to save space. Capacity as-a-service has become a business alternative for local data storage due to its low startup costs, low maintenance costs, and universal access to data regardless of location or device. Despite cost savings, availability, simplicity of use, adjusting, and sharing, it poses security risks as data is at the control of the cloud provider (CSP). Because of programming/equipment incapacity, it can mislead about information misfortune and debasement. Check the ownership of distributed storage information.
Traditional cryptographic solutions for data trustworthiness either need a local copy of the data (which data users (DUs) don't have) or allow DUs to download the entire data. The first arrangement demands more capacity, whereas the second increases document transport costs. To overcome this issue, several proposals use square less confirmation to evaluate trustworthiness without downloading all data. These works let the open verifier confirm, which is desirable. DUs can plan the assessing process with open review v. (TPA). It can convince CSP and DU. These proposals use proven information ownership (PDP) to guarantee ownership of information in unconfidential distributed storage by randomly confirming a few squares.
Recently, proposals have been made to allow TPA to verify cloud data's accuracy. Each plan has pros and cons. TPA shouldn't use the cloud server's response when inspecting. The plans in don't save lives. The processes provided in don't meet the information elements requirement, which allows information owners to embed, modify, and delete data without changing the metainformation of other blocks. Then, plans like couldn't meet clump checking requirement ensure that TPA can handle several concurrent check requests from DUs. This saves CSP and TPA computation and correspondence costs. Plans use blending-based cryptographic activities, which need extra time. We offer a safe and efficient information ownership protection scheme (SEPDP).
SEPDP helps information owners, group reviewing, and dynamic information duties. A probabilistic analysis of CSP's squares. We compared the proposed plan's exhibit to well-known systems.
The suggested plan's all-out check time is less than the present plan's. This means SEPDP can effectively test low-controlled devices. This paper's rest follows. Clarified elements prerequisites.

II. Literature Survey

A. Secure and constant cost public cloud storage auditing with deduplication

In order for cloud storage to be effective, data integrity and storage efficiency are two key needs. Data integrity for cloud storage is guaranteed by POR and PDP approaches. Storage efficiency is increased by POW, which safely deletes redundant data from the storage server. To accomplish both data integrity and storage efficiency, however, a minimal combination of the two strategies leads to non-trivial duplication of information (i.e., authentication tags), which is in opposition to POW's goals. Recent solutions to this issue have been shown to be insecure and to incur significant computational and communication costs. In order to offer effective and safe data integrity auditing together with storage deduplication for cloud storage, a new solution is required. In this study, we present a novel strategy for the solution of this open problem, based on homomorphic linear authenticators and polynomial-based authentication tags. Deduplication of files and the related authentication tags is possible thanks to our architecture. Storage deduplication and data integrity auditing are accomplished simultaneously.Constant real-time communication and computational expense on the user's end are further characteristics of our suggested approach. Both batch and public audits are supported.
As a result, our suggested method performs better than current POR and PDP schemes while incorporating deduplication as an additional utility.
We use the Computational Diffie-Hellman problem, the Static DiffieHellman problem, and the t-Strong Diffie-Hellman problem to demonstrate the security of our suggested system. Experimental findings on Amazon AWS and numerical analysis demonstrate how effective and scalable our system is.

B. Dupless: Server aided encryption for deduplicated storage AUTHORS: S. Keelveedhi, M. Bellare, and T. Ristenpart

Deduplication is a technique used by cloud storage service providers like Dropbox, Mozy, and others to store only one copy of each submitted file in order to conserve space. However, savings are lost if clients encrypt their files normally. This strain is alleviated by message-locked encryption, of which convergent encryption is the most notable example. However, brute-force assaults that can retrieve files that belong to a known set are fundamentally possible. We provide an architecture that offers safe deduplicated storage that can withstand brute-force attacks, and we implement it in the DupLESS system. Clients in DupLESS encrypt using

message-based keys that they have gotten from a key server via an unaware PRF protocol. It allows users to save encrypted data with a current service and have that service handle deduplication on their behalf.but still manages to secure adequate confidentiality assurances. We demonstrate how performance and space savings from encryption for deduplicated storage can be comparable to those of utilising the storage service with plaintext data.

III. Proposed System

In this paper, we propose two secure systems, SecCloud and SecCloud-D, in an effort to achieve data integrity and deduplication in the cloud.

By combining the management of a MapReduce cloud with an auditing entity, SecCloud enables its users to ensure the authenticity of data stored in the cloud and to generate metadata tags prior to upload.

SecCloud+ enables the guarantee of file confidentiality in addition to supporting integrity auditing and secure deduplication.

We propose an approach for performing direct audits of encrypted data's integrity.

A. Implementation

1. Cloud Service Provider

We create the Cloud Service Provider module in this module. This organisation offers a public cloud data storage service. The CS offers the data outsourcing service, stores data on behalf of the users, and uses deduplication to remove redundant data from storage and retain only unique information.

For the purposes of this research, we'll assume that CS is constantly online and has plenty of storage space and processing power.

2. Data Users Module

A user is a company that wishes to outsource data storage to the S-CSP and afterwards access the data.

In a storage system that allows for deduplication, the user only uploads one-of-a-kind data—which may belong to them or to other users— and does not upload any duplicate data to save on upload bandwidth.

At system setup, the authoriseddeduplication system grants a set of rights to each user. Each file is secured with a convergent encryption key and privilege keys to enable authoriseddeduplication with differential privileges.

B. Auditor

A MapReduce cloud is maintained by the auditor, which also serves as a certificate authority and assists clients with uploading and auditing their outsourced data. This presumption makes the auditor presumed to be connected to a set of public and private keys. It makes its public key visible to the other system entities. The capacity to validate the accuracy of data saved remotely is the primary design objective of this effort. public verification enables verification by anybody, not just the customers who originally stored the material.

4. Results And Discussion

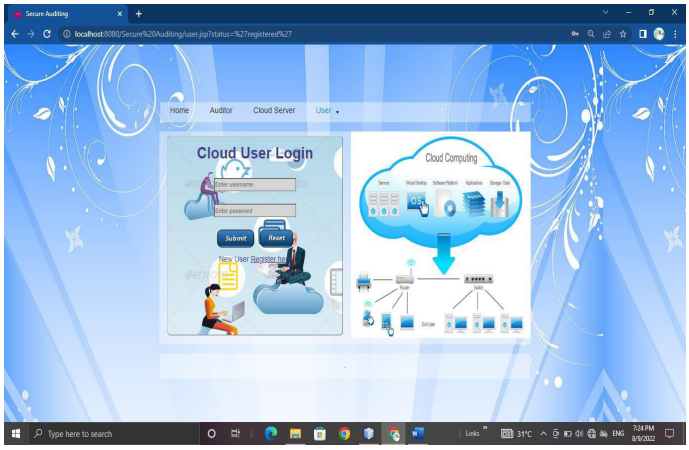

Fig 1 User login page

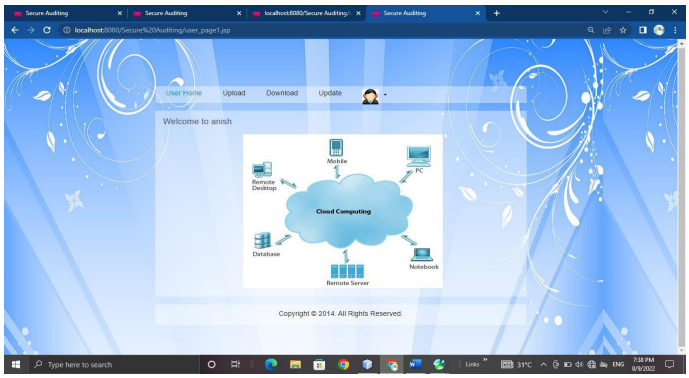

Fig 2: User home page

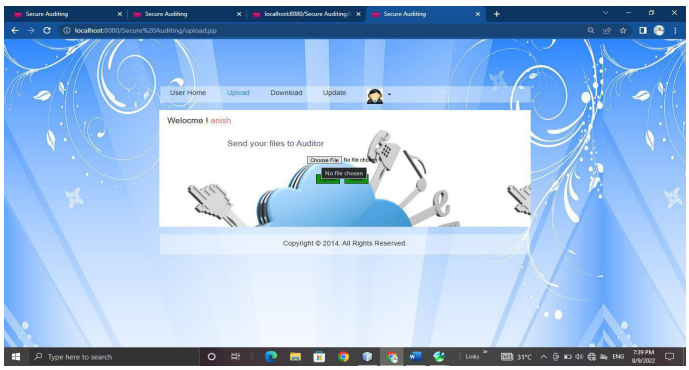

Fig 3: User file upload page

V. Conclusion

This work. LAST-HDFS allows you to load files. Independent of their In order to address the problem of physical location, you can store them cloud data placement control, we on the cloud based on policies. build a brand-new LAST-HDFS Additionally, it ensures that the system on top of the present HDFS in location policy is respected, even if data replication or load balancing have an impact on compliance with the policy. A robust LP-tree and Legal File Transfer graph were created to optimally distribute files with similar location preferences to the most suitable cloud nodes in order to increase the likelihood of spotting illegal file transfers. Both a large-scale simulated cloud environment and a real cloud testbed have been used by us for significant experimentation. Our team's experiments have shown that the LAST-HDFS system is both practical and effective.

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/u senixsecurity13/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.

[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] C. Erway, A. K¨upc¸ ¨u, C. Papamanthou, and R. Tamassia, ―Dynamic provable data possession,‖ in Proceedings of the 16th ACM Conference on Computer and Communications Security, ser. CCS '09. New York, NY, USA: ACM, 2009, pp. 213–222.

[8] F. Seb´e, J. DomingoFerrer, A. Martinez-Balleste, Y. Deswarte, and J.-J.Quisquater, ―Efficient remote data possession checking in critical information infrastructures,‖ IEEE Trans. on Knowl.and Data Eng., vol. 20, no. 8, pp. 1034–1038, 2008.