OPEN ACCESS INTERNATIONAL JOURNAL OF SCIENCE & ENGINEERING

LEVERAGING DE-DUPLICATION WITH SECURE AUDITING IN CLOUD DATA

Vidya Patil[1], Prof. Aruna Verma[2]
Dhole Patil College of Engineering, Pune, Maharashtra, India
patilvidya169@gmail.com[1], soni.aruna66@gmail.com[2]

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 de-duplication in cloud while achieving integrity auditing. This work, focuses the problem of integrity auditing and secure de-duplication on cloud data. Specifically, to achieve both data integrity and de-duplication in cloud, presented 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. 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.**

Keywords: Secure Data De-duplication, Encrypted Data, Data Availability, accountability, Cloud Computing.

I INTRODUCTION

Cloud storage is a model of internet enterprise storage where data is stored in virtualized pools of storage which is hosted by third-party. Cloud storage provides offers for customer which generated more benefit for cloud companies, like popularity, more user. Even though now days cloud storage system has been smart option for work. And also it is affordable, but it has certain limitation. The main problem of client data management and maintenance which is able to Relief by cloud server storage system of cloud is different from another storage System. The first problem is integrity auditing, i.e when we uploaded data it upload various manner like packets tokens which is less secure because if any packet loss while transmitting it's occur problem for client. As well as its to easy for a professional Attacker to attack. So its most important that maintain the integrity of data on storage system. The data is transferred via internet and stored in uncertain domain not the under control of client. The uncontrolled cloud server may passively hide the any problem related data for their reputation. It is more important that cloud server might even actively and deliberately discard rarely accessed data files belonging to an ordinary file. The second problem is secure deduplication. In cloud storage among these remote stored files, most of them are already on storage. According to recent survey by EMC, 70% of files are duplicated copies. Because its helps to cloud servers paid more for space from client. Thats the one of the reason why many cloud server are store duplicate copies of data. And Its more risky to available duplicate copies of data in storage.

Stored data is various manner like confidential password, banking detail, personal information, it is open invitation for attacker. In cloud server, server store every single file link with the who ask for the file. Cloud server needs to verify whether the user actually owns the file before creating a link for user. In de-duplicate data, when a user wants to upload a data file that already exists in the cloud storage, the cloud server executes a checking algorithm to see whether or not this user actually possesses the whole file i.e. it checks the file attribute. If the user passes the checking, he/she can directly use the file existed on the server without uploading it again. To overcome such problems cloud server uses proofs-of-ownership protocol, which let a client efficiently prove to a server that the client holds a file, rather than short information about it. In this a file has different ownership which introduce rigorous security definition. For working dynamic data proof-of-retrievability protocol used. Because dynamic data operation can be vital importance to storage outsourcing services.

II LITERATURE SURVEY

J. Li, X. Tan, X. Chen, and D. Wong proposed a new cloud storage architecture with two independent cloud servers, that is, the cloud storage server and the cloud audit server. The cloud audit server allows cloud users, to pre-process the data before uploading to the cloud storage server and verify data integrity. The cloud audit server eliminates the involvement of user in the auditing and in the pre-processing phases [6].

H.Wang, proposed the concept of PPDP. The systems proposed by author give its system model and security model and design an efficient pairing-based PPDP protocol. This PPDP protocol is provably secure and efficient by security analysis and performance analysis [5].

M. Bellare, S. Keelveedhi, and T. Ristenpart, formalize a new cryptographic primitive, Message-Locked Encryption (MLE), where the key under which encryption and decryption are performed is itself derived from the message. MLE provides a way to achieve secure de-duplication (space-efficient secure outsourced storage), a goal currently targeted by numerous cloud-storage providers[7].

M. Abadi, D. Boneh, I. Mironov, A. Raghunathan, and G. Segev, achieve the goal via a combination of a cut-and-choose technique and NIZKs. The resulting scheme is secure against a fully adaptive adversary. The second construction assumes a predetermined bound on the complexity of distributions specified by the adversary. It fits the original framework of deterministic MLE while satisfying a stronger security notion[8].

M. Azraoui, K. Elkhiyaoui, R. Molva, and M. Onen, presents StealthGuard, an efficient and provably secure proof of retrievabillity (POR) scheme. Stealth Guard makes use of a privacy preserving word search (WS) algorithm to search, as part of a POR query, for randomly valued blocks called watchdogs that are inserted in the file before outsourcing[9].

J. Li, X. Chen, M. i, J. Li, P. Lee, and W. Lou, implements Dekey using the Ramp secret sharing scheme and demonstrate that it incurs small encoding/decoding overhead compared to the network transmission overhead in the regular upload/download operations[10].

S. Keelveedhi, M. Bellare, and T. Ristenpart, presents a system, DupLESS, that combines a CE-type base MLE scheme with the ability to obtain message-derived keys with the help of a key server (KS) shared amongst a group of clients. The clients interact with the KS by a protocol for oblivious PRFs, ensuring that the KS can cryptographically mix in secret material to the per-message keys while learning nothing about files stored by clients[4].

III SYSTEM ARCHITECTURE

To solve this problem on existing system we present this secure system. Which generate better And Efficient system for accessing massive data on cloud. In this, firstly encrypted the plain data file and perform integrity auditing on that encrypted file. Division and Replication of Data in the Cloud for Optimal Performance and Security (DROPS) that judicially fragments user files into pieces and replicates them at strategic locations within the cloud. The division of a file into fragments is performed based on a given user criteria such that the individual fragments do not contain any meaningful information. Each of the cloud nodes (this system use the term node to represent computing, storage, physical, and virtual machines) contains a distinct fragment to increase the data security. A successful attack on a single node must not reveal the locations of other fragments within the cloud. To keep an attacker uncertain about the locations of the file fragments and to further improve the security.

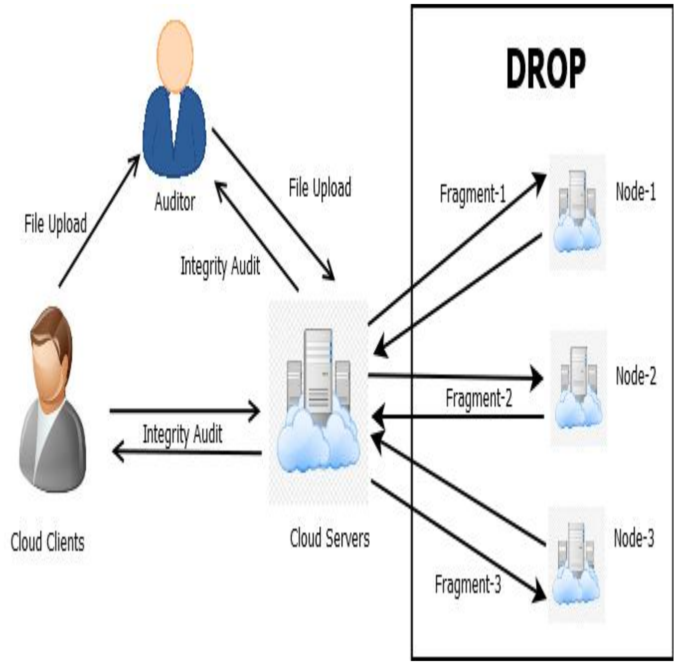


Figure 1 System architecture

3.1 Cloud Clients

Cloud Clients have large data files to be stored and rely on the cloud for data maintenance and computation. They can be either individual consumers or commercial organizations.

3.2 Cloud Servers

Cloud Servers virtualizes the resources according to the requirements of clients and expose them as storage pools. This system uses Division and Replication of Data in the Cloud for Optimal Performance and Security (DROPS) that fragments user files into pieces and replicates them at strategic node locations within the cloud. Typically, the cloud clients may buy or lease storage capacity from cloud servers, and store their individual data in these bought or rented spaces for future utilization.

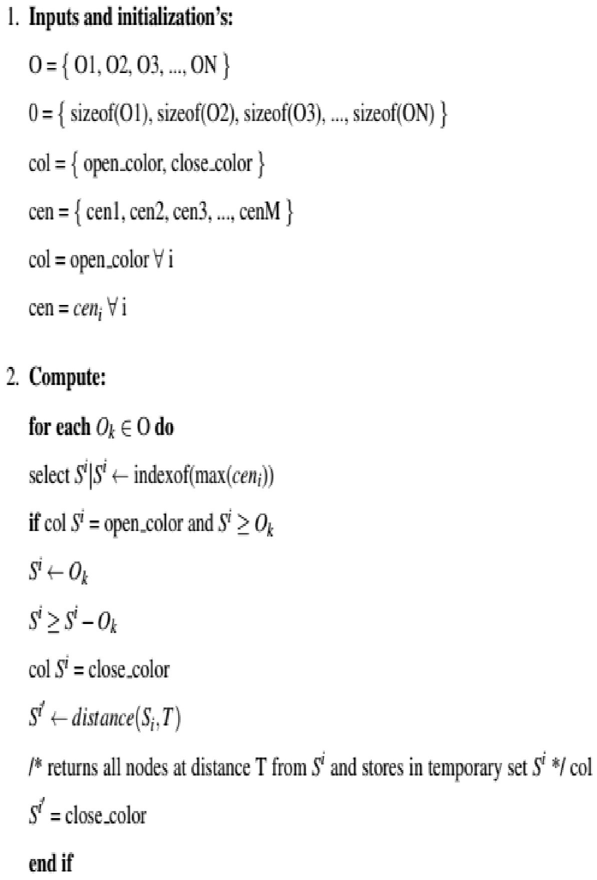
3.3 Auditors

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.

IV ALGORITHM

4.1 Fragment Placement Algorithm

This algorithm represents the fragment placement methodology. To deal with the security aspects of placing fragments, this system use the concept of T-coloring that was originally used or the channel assignment problem. It generate a non-negative random number and build the set T starting from zero to the generated random number. The set T is used to restrict the node selection to those nodes that are at hop-distances not belonging to T. If somehow the intruder compromises a node and obtains a fragment, then the location of the other fragments cannot be determined. The attacker can only keep on guessing the location of the other fragments



```
1. Inputs and initialization's:
   O = { O1, O2, O3, ..., ON }
   0 = { sizeof(O1), sizeof(O2), sizeof(O3), ..., sizeof(ON) }
   col = { open_color, close_color }
   cen = { cen1, cen2, cen3, ...., cenM }
   col = open_color ∀ i
   cen = $cen_i$ ∀ i

2. Compute:
   for each $O_k \in O$ do
   select $S^i | S^i \leftarrow$ indexof(max($cen_i$))
   if col $S^i$ = open_color and $S^i \geq O_k$
   $S^i \leftarrow O_k$
   $S^i \geq S^i - O_k$
   col $S^i$ = close_color
   $S^{i'} \leftarrow distance(S_i, T)$
   /* returns all nodes at distance T from $S^i$ and stores in temporary set $S^{i'}$ */ col
   $S^{i'}$ = close_color
   end if
```

V CONCLUSION AND FUTURE WORK

Aiming at achieving both data integrity and de-duplication in cloud, SecCloud and SecCloud+ is presented. 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. 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. DROPS technique can significantly reduce storage and bandwidth requirements by dividing a file into fragments, and replicate the fragmented data over the cloud nodes.

REFERENCES

1) Y. Zhu, H. Hu, G.-J. Ahn, and M. Yu, “Cooperative provable data possession for integrity verification in multicloud storage”, IEEE Transactions on Parallel and Distributed Systems, 2012.
2) J. Xu and E.-C. Chang, “Towards efficient proofs of retrievability”, in Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, 2012.
3) 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.
4) S. Keelveedhi, M. Bellare, and T. Ristenpart, “Dupless: Server aided encryption for deduplicated storage”, in Proceedings of the 22Nd USENIX Conference on Security, ser. SEC13. Washington, D.C.: USENIX Association, 2013.
5) H. Wang, “Proxy provable data possession in public clouds”, IEEE Transactions on Services Computing, vol. 6, no. 4, pp. 551559, 2013.
6) J. Li, X. Tan, X. Chen, and D. Wong, “An efficient proof of retrievability with public auditing in cloud computing”, in 5th International Conference on Intelligent Networking and Collaborative Systems (INCoS), 2013.
7) M. Bellare, S. Keelveedhi, and T. Ristenpart, “Message-locked encryption and secure deduplication”, in Advances in Cryptology EUROCRYPT 2013.
8) M. Abadi, D. Boneh, I. Mironov, A. Raghunathan, and G. Segev, “Messagelocked encryption for lock-dependent messages”, in Advances in Cryptology CRYPTO 2013.
9) M. Azraoui, K. Elkhiyaoui, R. Molva, and M. Onen, “Stealthguard: Proofs of retrievability with hidden watchdogs”, in Computer Security ESORICS 2014.
10) J. Li, X. Chen, M. Li, J. Li, P. Lee, andW. Lou, “Secure deduplication with efficient and reliable convergent key management”, IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 6, pp. 16151625, June 2014.