DATA DEDUPLICATION REDUCING AND DATA LEAKAGE IN MULTI CLOUD ENVIRONMENT

Mrs. B. Komali, M.C.A, Lecturer, Department of Computer Science, Sri DurgaMalleswara Siddhartha MahilaKalasala, Vijayawada.

Dr.K.Parish Venkata Kumar, M.Tech,Ph.D. Assistant. Professor,Department of Computer Applications, Velagapudi Ramakrishna Siddhartha Engineering College,Kanuru,Vijayawada.

Abstract— in this paper intuition is that outsourced data may require different levels of protection, depending on how popular it is: content shared by many users, such as a popular song or video, arguably requires less protection than a personal document, the copy of a payslip or the draft of an unsubmitted scientific paper. As more corporate and private users outsource their data to cloud storage providers, recent data breach incidents make end-to-end encryption an increasingly prominent requirement. Unfortunately, semantically secure encryption schemes render various cost-effective storage optimization techniques, such as data deduplication, ineffective. We present a novel idea that differentiates data according to their popularity. Based on this idea, we design an encryption scheme that guarantees semantic security for unpopular data and provides weaker security and better storage and bandwidth benefits for popular data. This way, data deduplication can be effective for popular data, whilst semantically secure encryption protects unpopular content. We show that our scheme is secure under the Symmetric External Decisional Diffe-Hellman Assumption in the random oracle model.

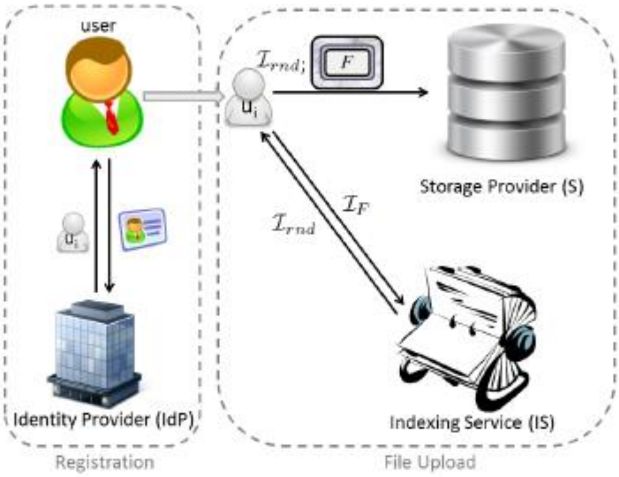
Keywords: privacy, Deduplication, Cloud Computing.

INTRODUCTION

SecCloud+ with multi-layered cryptosystem based Secure and Authorized Auditing deduplication model. In this paper, we present a scheme that permits a more fine-grained trade-off. Cloud computing is very difficult to audit the huge files and large amount of data. The first problem is integrity auditing. The cloud server is able to relieve clients from the heavy burden of storage management and maintenance. It provides the Integrity auditing by clustering the files with removing the duplicate files. 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 client's great concerns on the integrity of their data. The second problem is secure De-duplication. The duplicate files are mapped with a single copy of the file by mapping

DOI : 10.36893.DRSR.2021.V08I14.46-53

with the existing file in the cloud. 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. To overcome this, merkle hash tree function algorithm is used. This action of De-duplication 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).



LITERATURE REVIEW

Dupless: Server aided encryption for deduplicated storage

AUTHORS: S. Keelveedhi, M. Bellare, and T. Ristenpart

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.

DOI : 10.36893.DRSR.2021.V08I14.46-53

Secure and constant cost public cloud storage auditing with deduplication

AUTHORS: J. Yuan and S. Yu

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, 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.

Proofs of ownership in remote storage systems

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

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

DOI : 10.36893.DRSR.2021.V08I14.46-53

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.

Provable data possession at untrusted stores

AUTHORS: G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song

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. 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.

IMPLEMENTATION

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

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.

RELATED WORK

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. 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. The first problem is Integrity Auditing. The Cloud

Server is able to relieve clients from heavy burden of storage management without the control of the clients at all.The uncontrolled Cloud Servers may passively hide some data loss incidents from the clients. The second problem is Secure Deduplication. The Cloud Servers would like to deduplicate 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. This might lead to a number of threats potentially affecting the storage system. To overcome these existing problems this project proposes a system which handles both Integrity auditing and Data Deduplication. Hashing along encryption technique handles the problem of Data Integrity which ensures the safety of the client's storage in Cloud. Block-Level Deduplication (BLD) handles the problem of duplicate data storage, this helps in attaining an optimized Cloud storage for the clients.

ABE ALGORITHM:

The concept of **attribute based encryption** is a type of public-key encryption in which the secret key of a user and the ciphertext are dependent about attributes. In a system, the decryption of a cipher text is possible only if the set of attributes of the user key matches the attributes of the cipher text. A crucial security feature of Attribute-Based Encryption is collusion-resistance: An adversary that holds multiple keys should only be able to access data if at least one individual key grants access.

Step 1: Select File attribute1 – say File name

Step 2: Convert the file name to Binary Codes

Step 3: Select File attribute 2 – say file size

Step 4 : Convert the file size to Binary Codes

Step 5: Perform AND Operation of File Attribute 1 and 2

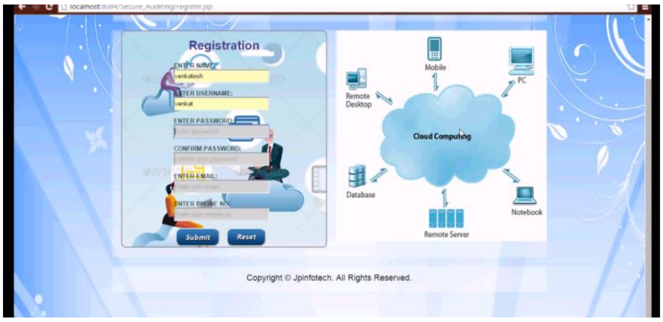
Step 6: Perform OR Operation of File Attribute 1 and 2

Step 7: Result of AND Operation Stored as Secret Key

Step 8: Result of OR Operation Stored as Public Key

DOI : 10.36893.DRSR.2021.V08I14.46-53

SAMPLE RESULTS

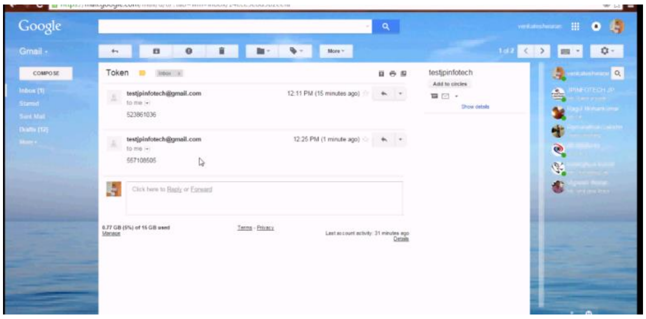


ENTER NAME: venkatesh

ENTER USERNAME: venkat

Copyright © Jpinfotech. All Rights Reserved.

Fig 1: Registration page



Token

testjpinfotech@gmail.com 12:11 PM (15 minutes ago)
to me
523861036

testjpinfotech@gmail.com 12:25 PM (1 minute ago)
to me
557108505

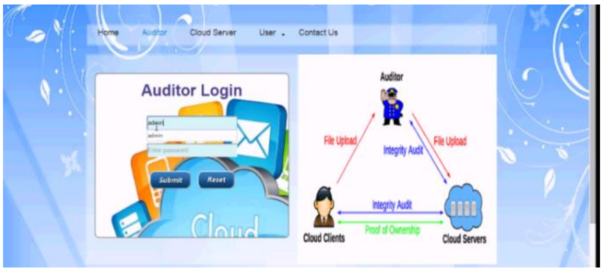
Click here to Reply or Forward

0.77 GB (5%) of 15 GB used
Manage

Terms - Privacy

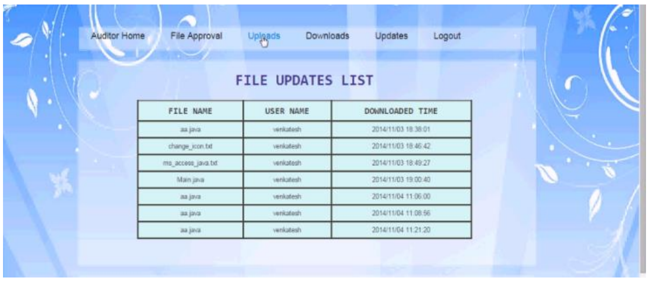
Last account activity: 31 minutes ago
Details

Fig 2: Security mail page



admin

Fig 3: Auditor login page



FILE UPDATES LIST

FILE NAME	USER NAME	DOWNLOADED TIME
aa.java	venkatesh	2014/11/03 18:38:01
change_icon.txt	venkatesh	2014/11/03 18:46:42
ms_access_java.txt	venkatesh	2014/11/03 18:49:27
Main.java	venkatesh	2014/11/03 19:00:40
aa.java	venkatesh	2014/11/04 11:06:00
aa.java	venkatesh	2014/11/04 11:08:56
aa.java	venkatesh	2014/11/04 11:21:20

Fig: Files update list page

DOI : 10.36893.DRSR.2021.V08I14.46-53

CONCLUSION

In this paper, 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.

In future have more scope to implement TPA (third party authority) for more security purpose.

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.
[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.
[8] 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.
[9] F. Seb´e, J. Domingo-Ferrer, 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.
[10] H. Wang, “Proxy provable data possession in public clouds,” *IEEE Transactions on Services Computing*, vol. 6, no. 4, pp. 551–559, 2018.

DOI : 10.36893.DRSR.2021.V08I14.46-53