IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 10, October 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.542

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 10, October 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0910021 |

Optimizing Information Leakage in Multi Cloud Storage Service

Bellary Shaik Yasmeen Banu, C.V. Madhusudan Reddy

PG Student, Dept. of C.S.E., Bheema Institute of Technology & Science, Adoni, India

Professor, Dept. of C.S.E., Bheema Institute of Technology & Science, Adoni, India

ABSTRACT: Many schemes have been recently advanced for storing data on multiple clouds. Distributing data over different cloud storage providers (CSPs) automatically provides users with a certain degree of information leakage control, for no single point of attack can leak all the information. However, unplanned distribution of data chunks can lead to high information disclosure even while using multiple clouds. In this paper, we study an important information leakage problem caused by unplanned data distribution in multicloud storage services. Then, we present StoreSim, an information leakage aware storage system in multicloud. StoreSim aims to store syntactically similar data on the same cloud, thus minimizing the user's information leakage across multiple clouds. We design an approximate algorithm to efficiently generate similarity-preserving signatures for data chunks based on MinHash and Bloom filter, and also design a function to compute the information leakage based on these signatures. Next, we present an effective storage plan generation algorithm based on clustering for distributing data chunks with minimal information leakage across multiple clouds. Finally, we evaluate our scheme using two real datasets from Wikipedia and GitHub. We show that our scheme can reduce the information leakage by up to 60% compared to unplanned placement. Furthermore, our analysis on system attachability demonstrates that our scheme makes attacks on information more complex.

KEYWORDS: Multicloud storage, information leakage, system attack ability, remote synchronization, distribution, and optimization.

I. INTRODUCTION

With the increasingly rapid uptake of devices such as laptops, cellphones and tablets, users require a ubiquitous and massive network storage to handle their ever-growing digital lives. To meet these demands, many cloud-based storage and file sharing services such as Dropbox, Google Drive and Amazon S3, have gained popularity due to the easy-to-use interface and low storage cost. However, these centralized cloud storage services are criticized for grabbing the control of users' data, which allows storage providers to run analytics for marketing and advertising. Also, the information in users' data can be leaked e.g., by means of malicious insiders, backdoors, bribe and coercion. One possible solution to reduce the risk of information leakages to employ multicloud storage systems in which no single point of attack can leak all the information. A malicious entity, such as the one revealed in recent attacks on privacy, would be required to coerce all the different CSPs on which a user might place her data, to get a complete picture of her data. Put simply, as the saying goes, do not put all the

eggs in one basket. Yet, the situation is not so simple. CSPs such as Dropbox, among many others, employ resync-like protocols to synchronize the local file to remote file in their centralized clouds [8]. Every local file is partitioned into small chunks and these chunks are hashed with fingerprinting algorithms such as SHA-1, MD5. Thus, a file's contents can be uniquely identified by this list of hashes. For each update of local file, only chunks with changed hashes will be uploaded to the cloud. This synchronization based on hashes is different from diff -like protocols that are based on comparing two versions of the same file line by line and can detect the exact updates and only upload these updates in a patch style Instead, the hash-based synchronization model needs to upload the whole chunks with changed hashes to the cloud. Thus, in the multicloud environment, two chunks differing only very slightly can be distributed to two different clouds. The following motivating example will show that if chunks of a user's data are assigned to different CSPs in an unplanned manner, the information leaked to each CSP can be higher than expected. Suppose that we have a storage service with three CSPs S1; S2; S3 and a user's dataset D. All the user's data will be firstly chunked and then uploaded to different clouds. The dataset D is represented as a set of hashes generated by each data chunk. This scenario is shown in Figure 1. In addition, we consider that the data chunks are distributed to different clouds in a round robin. Centralized cloud storage services are criticized for grabbing the control of users' data, which allows storage providers to run analytics for marketing and advertising. Also, the information in users' data can be leaked e.g.,

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 10, October 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0910021 |

by means of malicious insiders, backdoors, bribe and coercion. One possible solution to reduce the risk of information leakage is to employ multicloud storage systems in which no single point of attack can leak all the information.
Instead, the hash-based synchronization model needs to upload the whole chunks with changed hashes to the cloud. Thus, in the multicloud environment, two chunks differing only very slightly can be distributed to two different clouds. The following motivating example will show that if chunks of a user's data are assigned to different CSPs in an unplanned manner, the information leaked to each CSP can be higher than expected.

II. DISTRIBUTION AND OPTIMIZATION

Cloud storage services such as Dropbox and Google Drive,in essence, are centralized repositories for vast aggregations of personal data which can be monetized to afford the low cost(free) storage services for their users. While the users enjoy these storage services, they also lose their control on the data. Recent news about PRISM [6] shows that these CSPs can be compromised under coercion. Some other cloud storage services such as Wuala, Spider Oak employ client-side encryption to encrypt all the data before uploading the data. However, this does not change the inherent nature centralized architecture. As discussed previously, even with encryption, once the encryption key is exposed a user's entire data can be easily divulged

Disadvantages:

- Unplanned distribution of data chunks can lead to high information disclosure even while using multiple clouds.

- Frequent modifications of files by users result in large amount of similar chunks1.
- Similar chunks across files, due to which existing CSPs use the data de duplication technique.

III. PROPOSED SYSTEM

A. *Design Considerations:*

- We present StoreSim, an information leakage aware multicloud storage system which incorporates three important distributed entities and we also formulate information leakage optimization problem in multicloud.
- We propose an approximate algorithm, BFSMinHash, based on MinHash to generate similarity-preserving signatures for data chunks.
- Based on the information match measured by BFSMinHash, we develop an efficient storage plan generation algorithm, Clustering, for distributing user data to different clouds.

B. Description of the Proposed Algorithm:

In multicloud storage system, there are three distributed entities which synchronize users' data from the remote client to the cloud:
1) **Client** oversees pre-processing the users' data for the purpose of optimization, such as chunking (i.e., dividing files into individual chunks of a maximum size data unit), deduplication (i.e., avoiding storing and re-transmitting the same content already available on the remote servers), delta encoding (i.e., transmission of only modified portions of a file), bundling (i.e., the transmission of multiple small files as a single object) and encryption/decryption.
2) **Metadata servers** are used to store the metadata database about the information of files, CSPs and users, which usually are structured data representing the whole cloud file system.
3) **Storage servers** store the raw data blocks which can be both structured and unstructured data.

The most essential step of data synchronization is to detect updates. One solution is diff-like protocols which are based on comparing two versions of the same file line by line and can detect the exact updates. Only these updates will be uploaded to the cloud in a patch file which describes the difference between the old and the new version. However, diff-like protocols are not suitable for cloud storage services for three reasons. First, to compute the patch file, the client needs more storage overhead to store old versions, leading to the loss of users. Second, cloud storage services usually synchronize users' files across different clients and devices. If a file is modified in one client, then all other clients need to update both the old and the new version of this file, which results in high communication overhead. Finally, cloud storage services will be in great danger if the client bears the burden of maintaining revisionhistories. For example, a mistake of deleting old versions made by users can result in synchronization errors.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 10, October 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0910021 |

Advantages:

- However, previous works employed only a single cloud which has both compute and storage capacity. Our work is different since we consider a multi-cloud in which each storage cloud is only served as storage without the ability to compute.
- Our work is not alone in storing data with the adoption of multiple CSPs these works focused on different issues such as cost optimization, data consistency and availability.

Experimental Evaluation: first introduce the implementation of StoreSim, and the two datasets used for evaluation. Then we evaluate the performance of two algorithms, BFS Min-Hash and SPClustering. Finally, we analyze the time cost introduced by the leakage measure layer in StoreSim.
To Implement this, we have taken 3 modules.

1.**Client**: All operation related to client being performed. After Successfully registration User can login and upload the file in the cloud.

Fig:1. Client Registration with Valid Id **fig:1.2 Client Login**

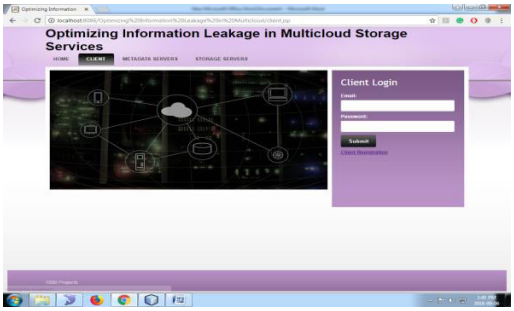


Fig.1.3: Upload Files: When Selecting the upload option user can upload the files into the cloud.

Fig:1.4: Split Files: After Uploading the files the data get split into chunks which is of 2 different files.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 10, October 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0910021 |

Fig:1.5. Modify Cloud1 Data: To check whether file is uploaded in cloud.

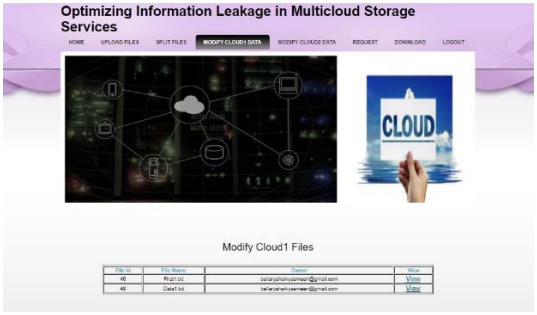
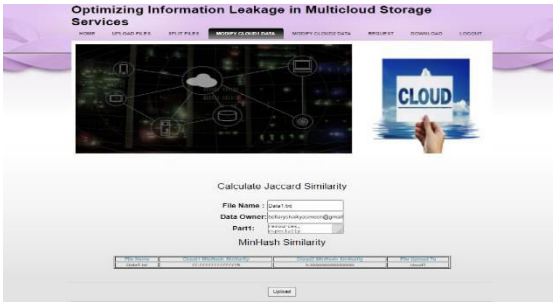


Fig:1.5.1: Selecting on View can show the information of the Data1.

Fig:1.5.2: Jaccard Similarity algorithm shows of any modification of file.



If any modification implemented on the data can be shown by the Jaccard Similarity algorithm. On clicking on submit updated data get uploaded in the cloud.

Same operations will be performed on Modify Cloud Data2.

Request files: to get the uploaded files we must request it. A request is sent to the storage server. In response to storage server a secret key is sent to registered email id.

Fig1.6. Requesting the server

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 10, October 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0910021 |

Download: Download option is used to download the file from Server. Initially that we need to pass the secret keys of both the files So that we can download both the chunks in single

Fig.1.7. Download Files

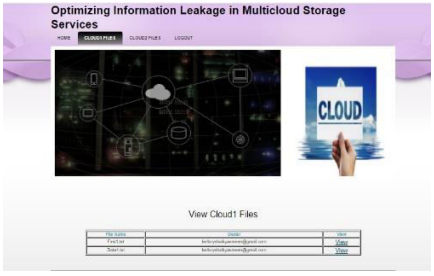
By selecting the download user can download the file from both the clouds. Only after the validation of secret keys sent by storage server.

Fig.1.7.1. Verification Secret Key.

2.Meta Data Server: are used to store the metadata database about the information of files, CSPs and users, which usually are structured data representing the whole cloud file system. Basically, its shows all the data that are stored in both the Clouds.

Fig:2.1 Meta Data server Home

Fig:2.2 View Cloud1 Files: OnView Selected shows **encrypted** data of file shows up.



| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 10, October 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0910021 |

3.Storage Server: User can keep track of all the files in servers. On Selecting the Cloud type either Storage Server1 (or) Storage Server2.

Fig:3.1 Storage Server1 Home

Fig:3.2 View Cloud1 Files

Fig:3.3 View Request:

Selecting the response send the secret key to the registered email. In Download Section of Client Module User can download the file by verifying the keys sent to mail id.Secret Key of Storage Server 1 and Secret Key of Storage Server 2 are sent to mail id .Both keys are mandatory to download the files .In download both files get combined and formatted into a single file which is ready to download.

IV. CONCLUSION

Distributing data on multiple clouds provides users with a certain degree of information leakage control in that no single cloud provider is privy to all the user's data. However, unplanned distribution of data chunks can lead to avoidable information leakage. We show that distributing data chunks in a round robin way can leak user's data as high as 80% of the total information with the increase in the number of data synchronization. To optimize the information leakage, we presented the StoreSim, an information leakage aware storage system in the multicloud. StoreSim achieves this goal by using novel algorithms, BFSMinHash and SPClustering, which place the data with minimal information leakage (based on similarity) on the same cloud. Through an extensive evaluation based on two real datasets, we demonstrate that StoreSim is both effective and efficient (in terms of time and storage space) in minimizing information leakage during the process of synchronization in multicloud. We show that our StoreSim can achieve near-optimal performance and reduce information leakage up to 60% compared to unplanned placement. Finally, through our attachability analysis, we further demonstrate that StoreSim not only reduces the risk of wholesale information leakage but also makes attacks on retail information much more complex.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 10, October 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0910021 |

REFERENCES

[1] P. Mahajan, S.Shetty, S. Lee, A. Clement, L. Alvisi, M. Dahlin, and M.Walfish, "Depot: Cloud storage with minimal trust," ACM Transactions on Computer Systems (TOCS), vol. 29, no. 4, p. 12, 2011.
[2] T.Suel and N.Memnon, "Algorithms for delta compression and remote file synchronization," 2002.
[3] I. Drago, E. Bocchi, M. Mellia, H. Slatman, and A. Pras, "Benchmarking personal cloud storage," in Proceedings of the 2013 conference on Internet measurement conference. ACM, 2013, pp. 205–212.
[4] I. Drago, M. Mellia, M. Munafo, A. Sperotto, R. Sadre, and A. Pras, "Inside drop box: understanding personal cloud storage services," in Proceedings ofthe 2012 ACM conference on Internet measurement conference. ACM, 2012, pp.481–494.
[5] A. Bessani, M. Correia, B. Quaresma, F. Andre's, and P. Sousa, "Dusky: dependable andsecure storage in a cloud-of-clouds," ACM Transactions on Storage (TOS), vol. 9, no. 4, p.12, 2013.
[6] H. Chen, Y. Hu, P. Lee, and Y. Tang, "Nc cloud: A network-coding-based storage systemin a cloud-of-clouds," 2013.

INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 7.542

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details