SECURE AND EFFICIENT DATA MIGRATION USING COUNTING BLOOM FILTERS IN CLOUD ENVIRONMENTS

[1] V.N.V.L.ABHISHIEK, [2]T.SUJILATHA, [3]R.M.MALLIKA

[1]PG Scholar, Dept. of CSE, Gokula Krishna College of Engineering, Sullurupet, AP, India.

[2]Asst. Professor, Dept. of CSE, Gokula Krishna College of Engineering, Sullurupet, AP, India.

[3]HOD, Dept. of CSE, Gokula Krishna College of Engineering, Sullurupet, AP, India.

Abstract – With the rapid development of cloud storage, an increasing number of data owners prefer to outsource their data to the cloud server, which can greatly reduce the local storage overhead. Because different cloud service providers offer distinct quality of data storage service, e.g., security, reliability, access speed and prices, cloud data transfer has become a fundamental requirement of the data owner to change the cloud service providers. Hence, how to securely migrate the data from one cloud to another and permanently delete the transferred data from the original cloud becomes a primary concern of data owners. To solve this problem, we construct a new counting Bloom filter-based scheme in this paper. The proposed scheme not only can achieve secure data transfer but also can realize permanent data deletion. Additionally, the proposed scheme can satisfy the public verifiability without requiring any trusted third party.

Index terms – Bloom Filter, cloud storage, public verifiability, Data deletion.

I. INTRODUCTION

Computing paradigm, connects large-scale distributed storage resources, computing resources and network bandwidths together [1,2]. By using these resources, it can provide tenants with plenty of high-quality cloud services. Due to the attractive advantages, the services (especially cloud storage service) have been widely applied [3,4], by which the resource-constraint data owners can outsource their data to the cloud server, which can greatly reduce the data owners' local storage overhead[5,6]. According to the report of Cisco[7], the number of Internet consumers will reach about 3.6 billion in 2019, and about 55 percent of them will employ cloud storage service.

Because of the promising market prospect, an increasing number of companies (e.g., Microsoft, Amazon, Alibaba) offer data owners cloud storage service with different prices, security, access speed, etc. To enjoy more suitable cloud storage service, the data owners might change the cloud storage service providers. Hence, they might

migrate their outsourced data from one cloud to another, and then delete the transferred data from the original cloud. According to Cisco[7], the cloud traffic is expected to be 95% of the total traffic by the end of2021, and almost 14% of the total cloud traffic will be the traffic between different cloud data centers. Foresee ably, the outsourced data transfer will become a fundamental requirement from the data owners' point of view.

To realize secure data migration, an outsourced data transfer app, Cloudsfer [8], has been designed utilizing cryptographic algorithm to prevent the data from privacy disclosure in the transfer phase. But there are still some security problems in processing the cloud data migration and deletion. Firstly, for saving network bandwidth, the cloud server might merely migrate part of the data, or even deliver some unrelated data to cheat the data owner[9]. Secondly, because of the network instability, some data blocks may lose during the transfer process.

Meanwhile, the adversary may destroy the transferred data blocks [10]. Hence, the transferred data may be polluted during the migration process. Last but not least, the original cloud server might maliciously reserve the transferred data for digging the implicit benefits. The data reservation is unexpected from the data owners' point of view. In short, the cloud storage service is economically attractive, but it inevitably suffers from some serious security challenges, specifically for the secure data transfer, integrity verification, verifiable deletion. These challenges, if not solved suitably, might prevent the public from accepting and employing cloud storage service.

II. BACKGROUND WORK

Verifiable data deletion has been well studied for a long time, resulting in many solutions. Xue et al.[19] studied the goal of secure data deletion, and put forward a key-policy attribute based encryption scheme, which can achieve data fine-grained access control and assured deletion. They reach data deletion by removing the attribute and use Merkle hash tree (MHT) to achieve verifiability, but their scheme requires a trusted authority. Du et al. designed a scheme called Associated deletion scheme for multi-copy (ADM), which uses pre-deleting sequence and MHT to achieve data integrity verification and provable deletion. However, their scheme also requires a TTP to manage the data keys. In 2018, Yang et al. presented a Blockchain-based cloud data deletion scheme, in which the cloud executes deletion operation and publishes the corresponding deletion evidence on Blockchain. Then any verifier can check the deletion result by verifying the deletion proof. Besides, they solve the bottleneck of requiring a TTP. Although these schemes all can achieve verifiable data deletion, they cannot realize secure data transfer.

To migrate the data from one cloud to another and delete the transferred data from the original cloud, many methods have been proposed. In 2015, Yu et al. presented a Provable data possession (PDP) scheme that can also support secure data

migration. To the best of our knowledge, their scheme is the first one to solve the data transfer between two clouds efficiently, but it's inefficient in data deletion process since they reach deletion by re-encrypting the transferred data, which requires the data owner to provide many information. Xue et al. designed a provable data migration scheme, which characterized by PDP and verifiable deletion. The data owner can check the data integrity through PDP protocol and verify the deletion result by Rank-based Merkle hash tree (RMHT). However, Liu et al. pointed out that there exists a security flaw in the scheme of Ref. and they designed an improved scheme that can solve the security flaw. In 2018, Yang et al. adopted vector commitment to design a new data transfer and deletion scheme, which offers the data owner the ability to verify the transfer and deletion results without any TTP. Moreover, their scheme can realize data integrity verification on the target cloud.

III. PROPOSED SYSTEM

1. Overview

In our scenario, we aim to achieve verifiable data transfer and deletion. The main processes are shown in Fig.1. Firstly, the data owner encrypts the data and outsources the ciphertext to the cloud A. Then he checks the storage result and deletes the local backup. Later, the data owner may change the cloud storage service provider and migrate some data from cloud A to cloud B. After that the data owner wants to check the transfer result. Finally, when the data transfer is successful, the data owner requires the cloud A to remove the transferred data and check the deletion result.

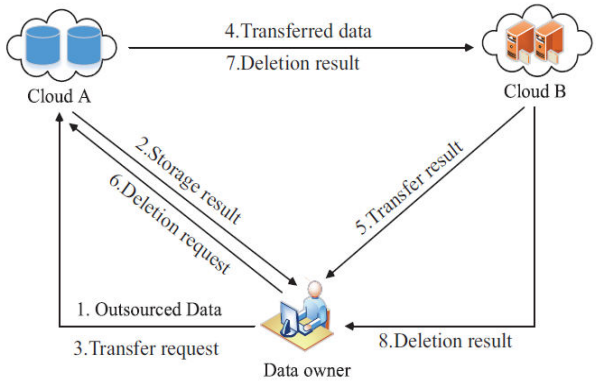

Fig. 1: the proposed System

2. The concrete scheme

Our new proposed scheme contains the following six algorithms.

1) Initialization

Generate ECDSA public private key pairs (PK_O, SK_O), (PK_A, SK_A) and (PK_B, SK_B) for the data owner, the cloud A and the cloud B, respectively. Then the data owner chooses k secure hash functions $g_1, g_2, \cdots, g_k$ that all map any integer in $[1,N]$ to distinct cells in CBF. Additionally, the data owner chooses a unique tag tag_f for the file that will be outsourced to the cloud A.

2) Data encryption

To protect the data confidentiality, the data owner uses secure encryption algorithm to encrypt the outsourced file before uploading.

3) Data outsourcing

The cloud A stores D and generates storage proof. Then the data owner checks the storage result and deletes the local backup.

4) Data transfer

When the data owner wants to change the service provider, he migrates some data blocks, even the whole file from the cloud A to the cloud B.

5) Transfer check

The cloud B wants to check the correctness of the transfer and returns the transfer result to the data owner.

6) Data deletion

The data owner might require the cloud A to delete some data blocks when they have been transferred to the cloud B successfully.

Implementation Modules

Multicloud: Lots of data centers are distributed around the world, and one region such as America, Asia, usually has several data centers belonging to the same or different cloud providers. So technically all the data centers can be access by a user in a certain region, but the user would experience different performance. The latency of some data centers is very low while that of some ones may be intolerable high. System chooses clouds for storing data from all the available clouds which meet the performance requirement, that is, they can offer acceptable throughput and latency when they are not in outage.

Owner Module: Owner module is to upload their files using some access policy. First they get the public key for particular upload file after getting this public key owner request the secret key for particular upload file. Using that secret key owner upload their file and performs Find all cost and memory Details, View Owner's VMs Details and purchase, Browse and enc file and upload, Check Data Integrity Proof, Transfer data from one to another cloud based on the price (Storage Mode Switching),Check all cloud VM details and Price list.

Cloud Storage: Cloud storage services have become increasingly popular. Because of the importance of privacy, many cloud storage encryption schemes have been proposed to protect data from those who do not have access. All such schemes assumed that cloud storage providers are safe and cannot be hacked; however, in practice, some authorities (i.e., coercers) may force cloud storage providers to reveal user secrets or confidential data on the cloud, thus altogether circumventing storage encryption schemes. In this paper, we present our design for a new cloud storage encryption scheme that enables cloud storage providers to create convincing fake user secrets to protect user privacy.

User Module: This module is used to help the client to search the file using the file id and file name. If the file id and name is incorrect means the user does not get the file, otherwise server ask the secret key and get the encryption file. If the user wants the decryption file means user have the secret key and performs View all attackers, View Resource Utilization Profiles (Total memory used for each and every data owner), View All VM and Price details, Resource Migration Check pointing (if it exceeds Threshold).

IV. RESULTS AND DISCUSSION

Data confidentiality

The data confidentiality means that adversary cannot get any plaintext information without the corresponding data decryption key. In our scheme, the data owner uses IND-CPA secure AES algorithm to encrypt the file. Meanwhile, the data decryption key is computed as k = H(tagf ‖SKO), where H is a secure hash function and SKO is the private key that kept secret. Hence, the adversary cannot forge a valid data decryption key successfully. Furthermore, the data owner keeps the data decryption key secret. That is, any adversary cannot obtain the decryption key to further get the plaintext information.

Data integrity

The data integrity means that the transferred data must be intact, or the cloud B refuses to accept the data. Upon receiving the transferred data (ai,Ci) from the cloud A and the hash values Hi from the data owner, the cloud B checks the equation Hi = H(tagf ‖ai‖Ci), where i ∈ ϕ. Note that {Hi}i∈ϕ are computed by the data owner with a secure hash function. Thus, the cloud A and other adversaries cannot forge a new data block (ai,C′ i) to make the equation Hi = H(tagf ‖ai‖C′ i) hold.

That is, if the cloud A does not honestly migrate the data to cloud B, or the transferred data blocks are tampered by the attackers during the migration process, the cloud B can detect these malicious behaviors and will not accept the received data. Hence, the integrity of the transferred data is guaranteed.

Public verifiability

We analyze the verifiability of the transfer result and the deletion result, respectively. The verifier who owns transfer proof π and transfer request Rt can verify the transfer result. Specifically, the verifier first checks the validity of Rt. If Rt is valid, it means that the data owner indeed requested to migrate the data to cloud B. Then the verifier further verifies the validity of the signatures sigta and sigtb. Note that the cloud B will not maliciously collude with the cloud A to mislead the data owner. Hence, the verifier can trust the returned transfer result if and only if both the signatures are valid. Besides, the verifier checks that whether the cloud B maintains the transferred data honestly by verifying the counting Bloom filter CBFb.

Experimental Setup

In this project we develop an web application java technology, database mysql and hardware requirement Dual core processor with 2 GB RAM. Using above technology we implement the our proposed method to secure data transfer and deletion.

Experimental Results

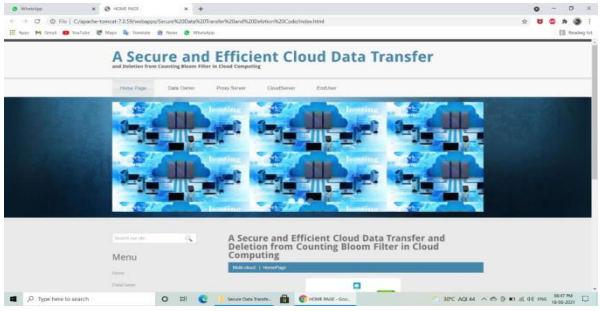

Fig. 2: Home Page

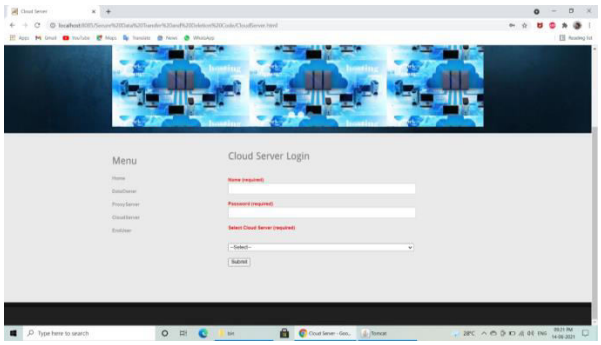

Fig. 3: Cloud Server Login

Fig. 4: Cloud Files

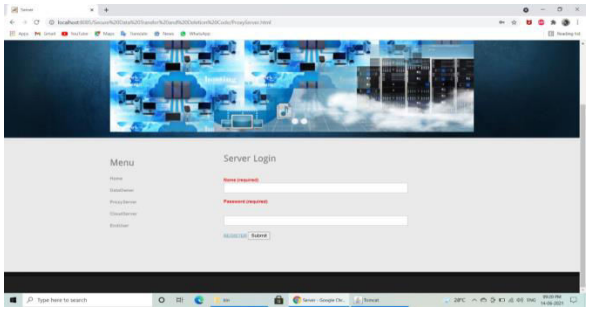

Fig. 5: Proxy Server Login

Fig. 6: Migrated File details

V. CONCLUSION

In cloud storage, the data owner does not believe that the cloud server might execute the data transfer and deletion operations honestly. To solve this problem, we propose a CBF-based secure data transfer scheme, which can also realize verifiable data deletion. In our scheme, the cloud B can check the transferred data integrity, which can guarantee the data is entirely migrated. Moreover, the cloud A should adopt CBF to generate deletion evidence after deletion, which will be used to verify the deletion result by the data owner. Hence, the cloud A cannot behave maliciously and cheat the data owner successfully.

REFRENCES

[1] C. Yang and J. Ye, “Secure and efficient fine-grained dataaccess control scheme in cloud computing”, Journal of HighSpeed Networks, Vol.21, No.4, pp.259–271, 2015.

[2] X. Chen, J. Li, J. Ma, et al., “New algorithms forsecure outsourcing of modular exponentiations”, IEEETransactions on Parallel and Distributed Systems, Vol.25,No.9, pp.2386–2396, 2014.

[3] P. Li, J. Li, Z. Huang, et al., “Privacy-preserving outsourcedclassification in cloud computing”, Cluster Computing, Vol.21,No.1, pp.277–286, 2018.

[4] B. Varghese and R. Buyya, “Next generation cloud computing:New trends and research directions”, Future GenerationComputer Systems, Vol.79, pp.849–861, 2018.

[5] W. Shen, J. Qin, J. Yu, et al., "Enabling identity-basedintegrity auditing and data sharing with sensitive informationhiding for secure cloud storage", IEEE Transactions on InformationForensics and Security, Vol.14, No.2, pp.331–346,2019.

[6] R. Kaur, I. Chana and J. Bhattacharya J, "Data deduplicationtechniques for efficient cloud storage management: Asystematic review", The Journal of Supercomputing, Vol.74,No.5, pp.2035–2085, 2018.

[7] Cisco, "Cisco global cloud index: Forecast and methodology,2014–2019", available at: https://www.cisco.com/c/en/us-/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.pdf, 2019-5-5.

[8] Cloudsfer, "Migrate & backup your files from any cloud to anycloud", available at: https://www.cloudsfer.com/, 2019-5-5.

[9] Y. Liu, S. Xiao, H. Wang, et al., "New provable datatransfer from provable data possession and deletion for securecloud storage", International Journal of Distributed SensorNetworks, Vol.15, No.4, pp.1–12, 2019.

[10] Y. Wang, X. Tao, J. Ni, et al., "Data integrity checking withreliable data transfer for secure cloud storage", InternationalJournal of Web and Grid Services, Vol.14, No.1, pp.106–121,2018.