EasyChair Preprint

№ 9295

Data Deletion Scheme Based on Bloom Filter.

Atla Naveen, Kokkonda Shiva and V Lavanya

November 10, 2022

DATA DELETION SCHEME BASED ON BLOOM FILTER

A Naveen , K Shiva
Department of Cloud Computing
School of Computing
SRM Institue of Science and Technology
Kattankulathur, India
an5867@srmist.edu.in
kg3734@srmist.edu.in

V. Lavanya
Department Of Networking And Communications,
Kattankulathur Campus,
SRM Institute of Science and Technology
Kattankulathur, India.

***Abstract* —** *One of the most alluring features of cloud computing is the ability to provide users limitless storage space. Users can therefore send their data to a cloud server, considerably reducing the need for local storage. However, because data ownership and management are separated in cloud storage, customers lose we cannot control the outgoing data, which presents numerous security and privacy issues. The issue of verifiable outsourced data erasure, this is significant yet has received little attention in business and academia, is the subject of this essay. The efficient fine-grained outsourced deletion of data we suggest is based on Bloom filter and produce storage and deletion results that are both publicly and privately verifiable.*

Key words: AWS, AWS LAMBDA, AWS S3, BLOOM FILTER.

INTRODUCTION

Cloud computing is a brand-new paradigm for computing that combines and develops parallel, distributed, and grid computing. One of the most alluring services provided by cloud computing is cloud storage, which can give customers easy access to data storage and business services by connecting a lot of distributed storage devices to a network. Users of cloud storage have the option to offload their data to the cloud server, considerably reducing the need for local hardware, software, and human resources. The widespread adoption of cloud storage in daily life and at work is a result of its alluring benefits.

As a result, more and more customers with limited resources, including people and businesses, prefer to use cloud storage services. If not properly addressed, these issues, particularly the one with data destruction, could prevent the general public from accepting cloud storage. Data deletion, the final stage life cycle of the data, directly impacts life cycle can conclude favourably, which is crucial for maintaining data security and privacy. Data integrity, on the other hand, has received much more attention because it has been well researched and solved. Although some verified deletion strategies for outsourced data in cloud computing environments have been presented, there are still some issues and concerns that must be decisively resolved right away.

First off, the majority of current methods are unable to provide fine-grained data deletion. In general, the information must be encrypted using a data key before being uploaded to the cloud server. In order to make the associated ciphertext unavailable, the data deletion can theoretically be accomplished by removing the matching data decryption key. However, the entire outsourced file will become inaccessible if the data decryption key is removed. The user typically wants to

erase a portion of the data in actual applications. In this instance, updating the entire outsourced file is required in order to achieve portion deletion, We create computational power and communication between user and cloud server . Designing fine-grained outsourced data deletion techniques that would allow the user to flexibly erase some unwanted data blocks is therefore desired.

LITERATURE

[1] The users update or store their data over cloud server, this reduces the need of local storage but as data is managed by cloud provider customer loses direct control or access over the deleting data, which leads to security and confidential issues, this is an important issue but most people neglect this , so therefore we suggest you to use data deletion using bloom filter and to produce results of amount of storage left and the data that is deleted. In this way the data which is deleted can be verified and integrity is maintained.

[2] As we know cloud helps users to store a lot of data. The uploaded data stores in big data center far from the trust zone of the owner who owns data, which may lead to the problem of data confidentiality. The stored data can be stolen in a digital way or physical way which may cause damage for data privacy. To avoid this we use SSGK, which means secret sharing group key management protocol which helps in encrypting the data and only the persons having the key can access the data. SSGK uses a group key to safe a file or data and to distribute the key they use secret sharing technique.

[3] We consider multi-copy somewhat ineffective in this kind of situation because many PDP protocols depend on public key infrastructure(PKI) which has many security flaws and very high costs on communication and computation, hence we suggest PDP strategy on multicopy among different servers for improving efficiency and security over cloud which helps and promotes its clients to store data over cloud , the verifiable tags can be used in maintaining over all copies.

[4] We initially examine the objectives of ensured data erasure in this study before formalising its security architecture.Then, they suggested for an assured deletion and key policy ABE schemev(AD-KP) for the data. For pure data deletion, the architecture used the Merkle Hash Tree and the attribute revocation cryptographic primitive. The suggested AD-KP-ABE has advantageous characteristics including ensured data erasure, partial ciphertext updating, and no secret key update. The thorough security proof and implementation outcomes show that our solution is secure and workable.

[5] The group manager will update all users' private keys when they depart, with the exception of those whose access has been revoked. Additionally, because the computation cost for the CP-ABE system increases linearly with access structure complexity, it has a high cost. We outsource heavy calculation workloads to cloud service providers in order to lower computation costs while preventing the disclosure of file content and secret keys. Notably, our system is users who work together with active users. Under the Diffie-Hellman assumption of divisible computing, we demonstrate the security of our system. Our experiment's findings demonstrate that local devices can have relatively low and consistent processing costs. Devices with limited resources can use our method.

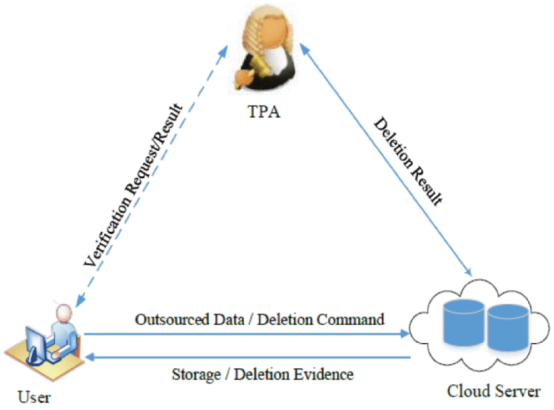
METHODOLOGY:

A)SYSTEM MODEL:

The entities can be divided as user,server and third party auditor, the user decides the data with need to be uploaded over the cloud by saving local storage and computation

,then later if the user decides if some amount of data isn't required the user can delete the data. As the cloud server has more computational power and big storage capacity, it gives us convenient methods for data storage

and deletion of data. The third party advisor(TPA) has mutual trust in both sides by user and cloud server, he solves the disagreements between user and cloud server.



B)SECURITY CHALLENGES:

Talking about the security we need to consider three challenges.

1)DATA SECURITY:

Cloud provides storage for storing data, many users store their vast data in cloud which may lead to data privacy disclosure. The data can be stolen either the way from internal (cloud server) or from external(hackers) which leads to data disclosure. Not only the cloud server and hackers are the reason for data loss but also one of the reason can be failure of software and hardware.

2)DISHONEST BEHAVIOR:

Cloud server may not delete the data properly it may search for valuable resources or information in our data by using some malicious methods , in the other way data deletion might cost us plenty of overhead resources as cloud isn't trusted here.

3)CONFLICTS:

In this the both side might create conflicts depending on their behavior and might blame each other , where the user may deny the aquisations on the deletion command cloud deleting the data without notice and the cloud server might retain the users data after deletion through malicious terms or methods.

C) DESIGN GOALS:

These can be divided into three parts.

1)CONFIDENTIALITY OF THE DATA:

As we preserve information by keeping secret. We can use IND-CPA secure encryption on our data before storing in the cloud, without the decryption key they cannot open our encrypted data and integrity of our data is maintained.

2)DATA DELETION:

The data deletion is done when a malicious behavior appeared or the file is kept idle for long time. The records of the deletion can be seen in logs.

3)NON REPUDIATION:

For guaranteeing that participants do not blame each other , if the dishonest participant blames the honest one by behaving maliciously the honest one must be able to detect and must be able to provide some proof to prove that the other participant is guilty

FLOW OF ACTIVITY:

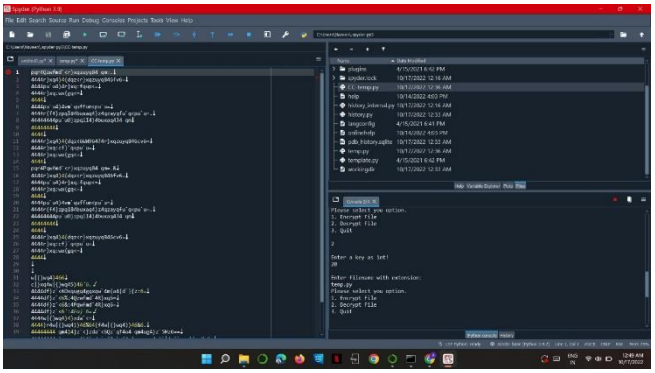
A)IMPLEMENTATION:

We study the drawbacks in security and integrity of data deletion in cloud and to solve this problem we propose the usage of IBF(invertible bloom filter) for data deletion data is split into different blocks before deletion, in result the user requires the cloud server to erase some unnecessary blocks , the server must override the unnecessary blocks and remove them using IBF, now the user can able to check the result and data which is deleted by cloud by relating if the index belongs to the IBF or not in IBF, here the randomly inserted blocks are not deleted which guarantees the deleted data isn't empty, the cloud server always generates the evidences regarding data deletion, even if everything is deleted.

1)ENCRYPTION:

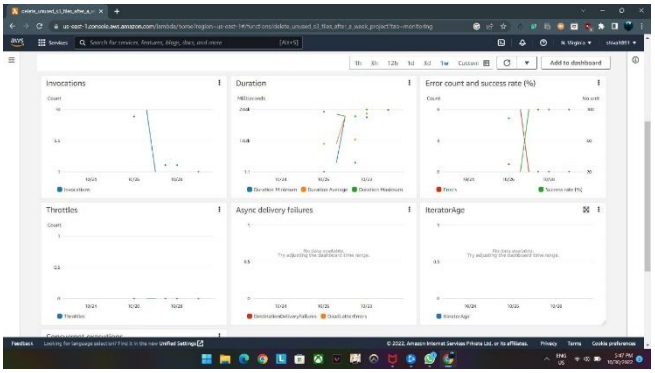
We open the file then read the data in file and store the data and close the file and later then we take an input key and the binary address of the data is changed by apply xor function between key and the

actual binary address now the binary address is changed and uploaded and later if we want to decrypt use the same key and the binary address of the uploaded file and perform xor function to decrypt the file.



2)MONITORING:

Here the logs of the accessed the buckets are generated we get the details of the users who access our bucket and any changes made are observed, this helps us to maintain integrity and confidentiality of the file.

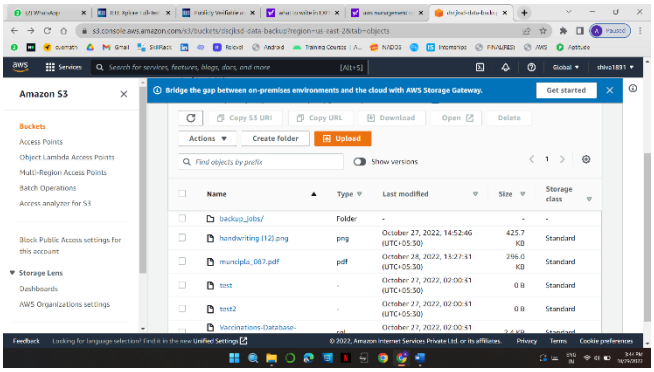


3)DELETION:

The file uploaded in s3 will be monitored and if the count of the file accessed is grater than the given or specified count then the file automatically replicates to another s3 bucket and the file gets deleted the main bucket. If the file is also idle for a specific time then also the file is replicated and deleted.

4)VERIFICATION:

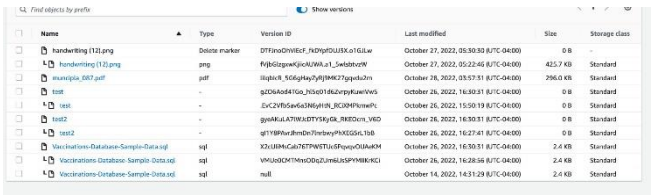
The deleted proofs can be seen in the s3 logs and using AWS SNS we can get notifications for the actions performed.

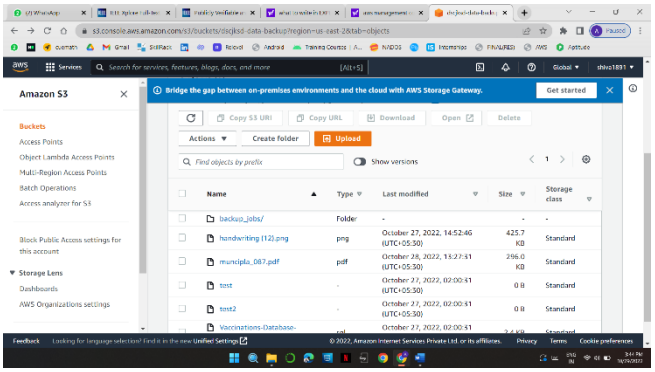


RESULTS AND DISCUSSIONS :

Firstly we create a bucket in 2 or more buckets in s3 and then we replicate our data in various regions and include some security policies, then later we add trigger if a using cloud watch if the trigger is activated then the the file gets deleted in the original bucket and is backed up by the other bucket in different region, the conditions for the trigger to get activated is the file must be opened multiple times within a particular time period.

Drawbacks of the data collected is that, a lot of dummy data is put to used which might not effect the real data which is collected by the user, In future we may add notification for priority of events and give more detailed information. We also plan to examine the connections existed between real and predicted performance depending on the actual data collected from platform by using an application.





REFERENCES

[1] Yang, C., Tao, X., Zhao, F. and Wang, Y., 2020. Secure data transfer and deletion from counting bloom filter in cloud computing. *Chinese Journal of Electronics*, *29*(2), pp.273-280.

[2] Du, Y., He, G. and Yu, D., 2016, August. Efficient hashing technique based on bloom filter for high-speed network. In *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)* (Vol. 1, pp. 58-63). IEEE.

[3] Yan, Y., Wu, L., Gao, G., Wang, H. and Xu, W., 2018. A dynamic integrity verification scheme of cloud storage data based on lattice and Bloom filter. *Journal of information security and applications*, *39*, pp.10-18.

[4] Guo, C., Zhuang, R., Chang, C.C. and Yuan, Q., 2019. Dynamic multi-keyword ranked search based on bloom filter over encrypted cloud data. *iEEE Access*, *7*, pp.35826-35837.

[5] Y. Wang, X. Tao, J. Ni, and Y. Yu, ‘‘Data integrity checking with reliable data transfer for secure cloud storage,’’ Int. J. Web Grid Services, vol. 14, no. 1, pp. 106–121, 2018, doi: 10.1504/IJWGS.2018.088396.

[6] C. Yang, X. Tao, and F. Zhao, ‘‘Publicly verifiable data transfer and deletion scheme for cloud storage,’’ Int. J. Distrib. Sensor Netw., vol. 15, no. 10, pp. 1–12, 2019, doi: 10.1177/1550147719878999

[7] Y. Liu, S. Xiao, H. Wang, and X. Wang, ‘‘New provable data transfer from provable data possession and deletion for secure cloud storage,’’ Int. J. Distrib. Sensor Netw., vol. 15, no. 4, pp. 1–12, 2019, doi: 10.1177/1550147719842493.

[8] L. Xue, J. Ni, Y. Li, and J. Shen, ‘‘Provable data transfer from provable data possession and deletion in cloud storage,’’ Comput. Standards Interfaces, vol. 54, pp. 46–54, Nov. 2017, doi: 10.1016/j.csi.2016.08.006.

[9] C. Yang, Q. Chen, and Y. Liu, ‘‘Fine-grained outsourced data deletion scheme in cloud computing,’’ Int. J. Electron. Inf. Eng., vol. 11, no. 2, pp. 81–98, 2019, doi: 10.6636/IJEIE.201912_11(2).04.

[10] B. Yang, ‘‘Efficient attributebased encryption with attribute revocation for assured data deletion,’’ Inf. Sci., vol. 479, pp. 640–650, Apr. 2019, doi: 10.1016/j.ins.2018.02.015.

[11] Pagh, A., Pagh, R. and Rao, S.S., 2005, January. An optimal bloom filter replacement. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms* (pp. 823-829).

[12] Yang, C., Tao, X., Zhao, F. and Wang, Y., 2019, June. A new outsourced data deletion scheme with public verifiability. In *International Conference on Wireless Algorithms, Systems, and Applications* (pp. 631-638). Springer, Cham.

[13] Marandi, A., Braun, T., Salamatian, K. and Thomos, N., 2017, June. BFR: a bloom filter-based routing approach for information-centric networks. In *2017 IFIP Networking Conference (IFIP Networking) and Workshops* (pp. 1-9). IEEE.

[14] Sbarski, P. and Kroonenburg, S., 2017. *Serverless architectures on AWS: with examples using Aws Lambda*. Simon and Schuster.

[15] Geravand, S. and Ahmadi, M., 2013. Bloom filter applications in network security: A state-of-the-art survey. *Computer Networks*, *57*(18), pp.4047-4064.