*Cloud Security Using the Smart Contracts*

**1Veena S, 2Tejas Mallikarjun C J,3Vishwesh Adiga S, 4Venkatesh Reddy B C, 5Yogish P D**

**1,2,3,4,5Department of Electronics and Communication Engineering**

**1, 2,3,4,5PES University, Bangalore, Karnataka, India**

[**veenas@pes.edu**](mailto:veenas@pes.edu)**,** [**tejasmallikarjuncj@pesu.pes.edu**](mailto:tejasmallikarjuncj@pesu.pes.edu)**,** [**venkateshreddybc@pesu.pes.edu**](mailto:venkateshreddybc@pesu.pes.edu), [**vishweshadiga@pesu.pes.edu**](mailto:vishweshadiga@pesu.pes.edu)**,** [**yogishpd@pesu.pes.edu**](mailto:yogishpd@pesu.pes.edu)**,**

***Abstract— In today's digital age, it's becoming increasingly common for individuals and businesses alike to opt for cloud-based storage over traditional on-premise storage. However, this shift towards cloud computing also brings with it a host of new security challenges. Data stored on public cloud infrastructure is susceptible to various types of vulnerabilities, threats, and attacks that can compromise its confidentiality, availability, and integrity.***

***To address these concerns, we propose an innovative architecture that integrates multiple entities - including multiple public clouds, a private cloud, smart contracts, and the data owner, client, or user. Our decentralized cloud architecture provides a robust form of security for data by incorporating smart contracts between the data owner and the private cloud, as well as between the client and the data owner.***

***By utilizing this approach, we can help ensure that data is protected from unauthorized access, while also providing users with a highly flexible and scalable solution for managing their data in the cloud. With our architecture, businesses and individuals can enjoy the benefits of cloud-based storage while also maintaining the highest levels of security and peace of mind.***

***Keywords-*** ***cloud, smart contract. AONT, data owner, client***

# INTRODUCTION

The use of public cloud for storing the data is not making the issue of security any easier as public clouds are not safe when compared to their private counterparts. Public clouds can be hacked or be easily accessed by another unauthorized person or an entity(enterprise/organization). The current security standards do provide robust forms of security but none of them were able to fully secure the cloud infrastructure. This motivated us to come up with a distributed blockchain based cloud architecture where we use smart contracts along with AONT techniques to alleviate the existing safety and security measures.

When an individual’s or an organization’s data is stored on public cloud, which helps in cutting costs to a greater extent. The data as we know, it is devolved through the internet layers(i.e., TCP/IP model or OSI Model), the data could be compromised or deleted or changed i.e., it refers to the confidentiality, integrity, availability of our data.

And also, the data is subject to many kinds of attacks such ad Denial of service (DoS) attacks, forking attacks, Man in the Middle attack, exhaustive key attacks etc. To control these types of attacks many forms of security measures or protection means such as authentication, authorization, access control list is present. The architecture proposed by us includes the following entities i.e., the private cloud, many public clouds, the smart contracts, the clients and the data owner.

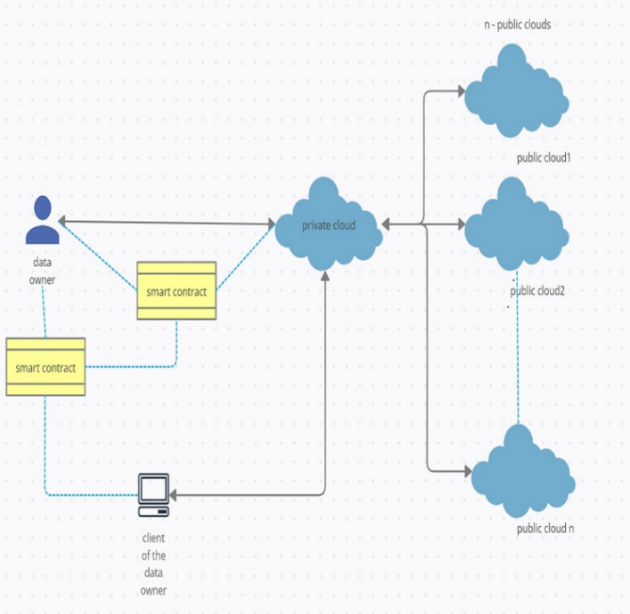
An internal cloud is another term for a private cloud. Instead of the wider public, a private cloud offers computer services to a small group of users and a private internal network (inside the enterprise). Internal hosting and firewalls ensure the privacy and security of data in the private cloud. It also ensures that untrusted parties are unable to access crucial or confidential data. The public cloud is a shared platform available to anybody with an Internet connection. The public cloud is managed by a third party, specifically a cloud service provider, on a pay-as-you-go basis. The public cloud's shared storage is being used concurrently by many users. A Data Owner is the person in charge of one or more data sets in an organization's classification, preservation, utilization, and quality. This requirement includes, but is not limited to, ensuring that the organization's Data Glossary is comprehensive and that all stakeholders have agreed on it. Client is the entity who uses or utilizes our architecture, the client can be an individual person or an organization/enterprise/company. Like any other contract, a smart contract outlines the conditions of an agreement. Unlike a conventional contract, a smart contract's terms are implemented as code that runs on a blockchain like Ethereum. Smart contracts can be used by programmers to create apps that take use of the security, dependability, and accessibility of the blockchain while providing complex peer-to-peer functionality.

# LITERATURE SURVEY

As a reference we can see the various researches done by other researchers and authors who have focused on pubic cloud security for mitigating different types of attacks pertaining to CIA (Confidentiality, Integrity, Availability) of the data.

M. Shah and his team [1] explained about the IPFS (Inter-Planetary File System) protocol, which is used in the proposed system to encrypt and distribute the user's file among several network peers. The IPFS generates hashes. The route to the file is identified by the hash value, which is stored on the blockchain. This study focuses on secure data storage that is decentralized, high data availability, and efficient storage resource use.

S. G. Sharma and his team [2] introduces blockchain and contrasts it with the various platforms it may be applied to. Examples of blockchain applications that could be utilized to build a secure cloud computing infrastructure are given in this paper.

E. Bacis, S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, M. Rosa and P. Samarati [3] suggested a storage strategy for resource owners that makes use of decentralized cloud services to effectively store and securely delete their resources. Their method combines All-Or-Nothing-Transform with thoughtful resource slicing algorithms and its distributed decentralized storage network for reliable resource preservation. V. Amrutiya, S. Jhamb, P. Priyadarshi and A. Bhatia [4] suggested a solution for two-factor authentication without calling for the creation and distribution of authentication tokens to users by a centralized, trusted third party

The Galois/Counter Mode of Operation (GCM) [5] is a white paper that was published that provides a comprehensive explanation of the GCM mode of operation and suggests utilizing hardware to achieve high speeds at a low cost and low latency. GCM also possesses additional advantages. When there is no data to encrypt, it can function as a standalone MAC and authenticate communications without any changes. It also acts as an incremental MAC because an authentication tag can be computed for a new message with a computational cost proportional to the number of changed bits if a portion of a message is changed after an authentication tag has been computed for that message. This particular feature is not present in any of the modes that have been suggested. X. Shen and Z. Li [7] elaborates about the ability of a company to obtain accurate and timely data about its operations, manage this data efficiently, and use that data to analyze and guide its activities is more important than ever. Explained about the importance of using a DBMS to manage data has a variety of advantages over traditional file management systems, including data independence, effective data access, concurrent access, crash recovery, data administration, data integrity, and security.

# PROPOSED MODEL

Our proposed architecture consists of a data owner, a private and public cloud provider, and a client for accessing the data. The unique novelty of our solution lies in its advanced use of AONT (All or Nothing Transformation) to fragment the data, which is then stored randomly across multiple public clouds by the private cloud. This fragmentation technique, combined with the use of a mapping table, ensures that the data is highly secure and resistant to attacks.

Furthermore, our model employs a sophisticated system of smart contracts to manage key access and ensure maximum security. Specifically, we have utilized blockchain-based smart contracts to control user and file access, providing a highly efficient and secure solution for businesses and individuals seeking to leverage the power of the cloud.

Overall, our architecture represents a significant breakthrough in data security and management, offering a robust and reliable solution for data storage and management in the cloud. With its advanced fragmentation techniques, advanced smart contract management, and sophisticated use of blockchain technology, our model is poised to revolutionize the way we think about data security and management in the cloud.

Fig 1. Proposed model block diagram

# IMPLEMENTATION

We have proposed a model using Python Flask as a backend API in the cloud, and a command-line interface as a frontend. To emulate a local blockchain, we have used the Remix IDE, the Brownie Python module, and the Ganache CLI (command-line interface). For our database, we have used MySQL. However, since we used a command-line interface, we did not include a loopback function for when users enter invalid details.

## New user signup procedure

When new user approaches the private cloud, private cloud asks for username, Ethereum id, password and if he is a data owner. After that private cloud checks whether user already presents, if not user will be added. Then private cloud checks if the user is a data owner or not. If user is a data owner, then the private cloud will create smart contract between data owner and itself

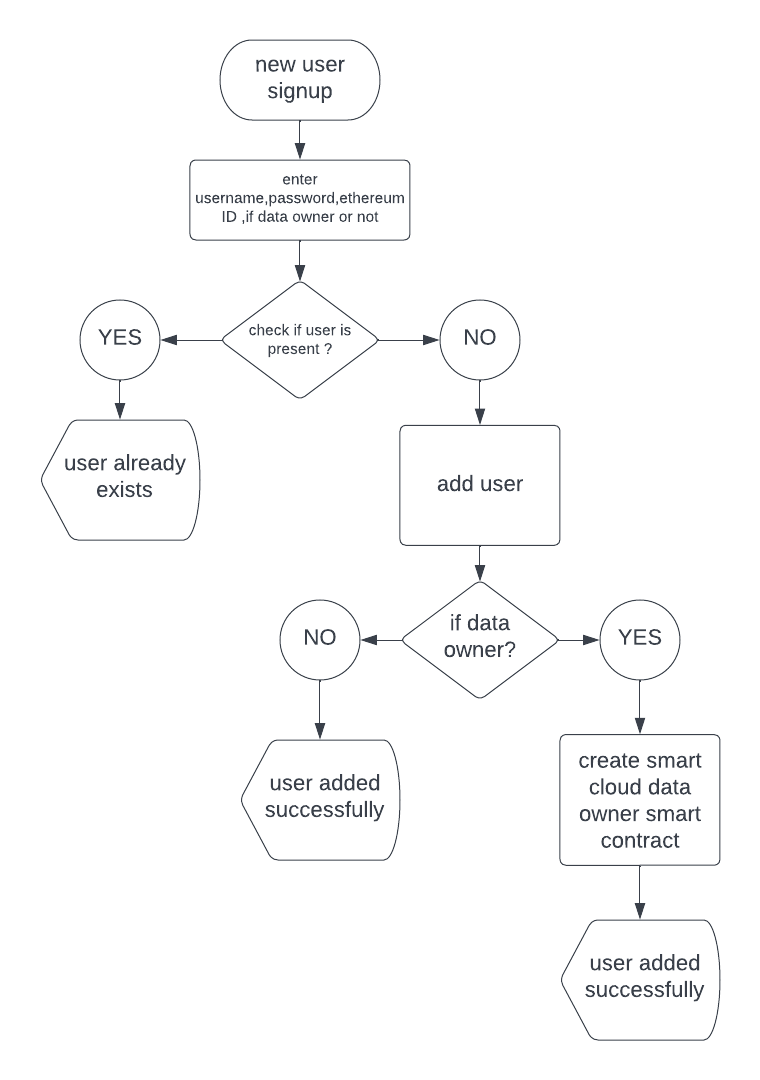


Fig 2. New user signup procedure flowchart

## B. Data storage procedure

When the owner of the data decides to store the data in the public cloud, the owner of the data will transfer all of his or her data to the private cloud. After the user's username and password have been verified, the private cloud checks to see if the file name is already present. The file will undergo AONT transformation and AES encryption in GCM mode if it was not present previously.

After all this processes n chunks of data will be stored in public clouds randomly. Along with this private also creates mapping table which contains chunk index, cloud id and key tag and sends the mapping table to the data owner for security purpose.

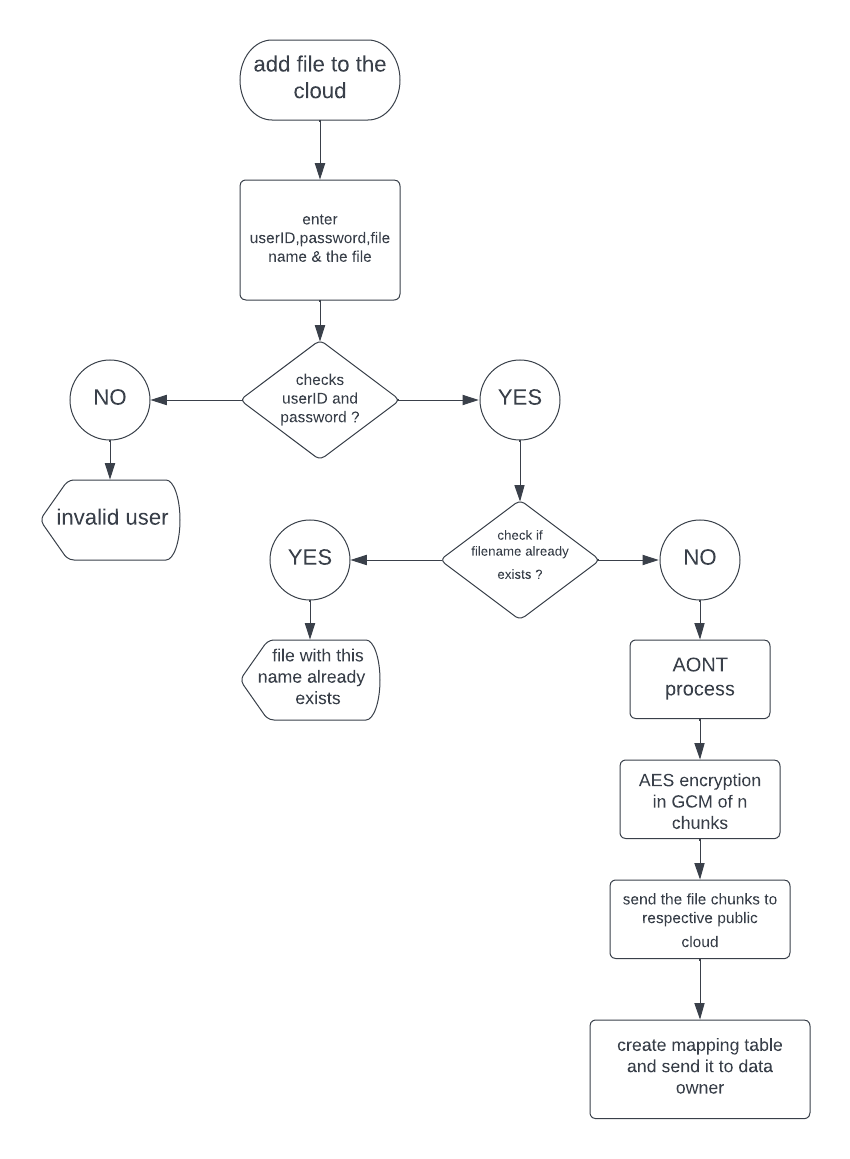


Fig 3. Data storage procedure flowchart

## a) Private cloud working while storing the data

Initially the private cloud divides the data into N chunks.Each data chunk is then encrypted using symmetric key encryption method i.e, AES (Advanced Encryption Standard) mode.After that each chunk of data will be hashed.The N Hashed chunks are xored with their respective indices.

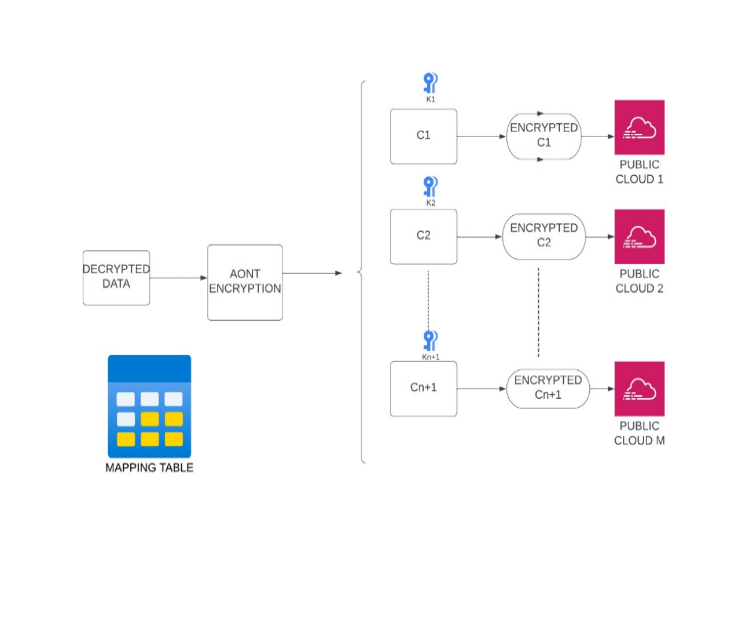


Fig 4. Private cloud working block diagram while storing data

The key is then taken and xored with all the hashed xored chunks. It results in (N+1)th chunk of the data. Now all N+1 chunks will be encrypted using AES in GCM mode.Now, the pre-processed N chunks will be randomly sent to different public clouds but (N+1)th chunk will be retained in the private cloud itself for security purposes.After that each chunk of data will be hashed.

## C. Adding client

When data owner approaches the private cloud for adding new clients, private cloud asks for username, Ethereum id, password. Then private cloud checks if the he is a data owner or not. If he is a data owner, then the private cloud will add new client to the smart contract.

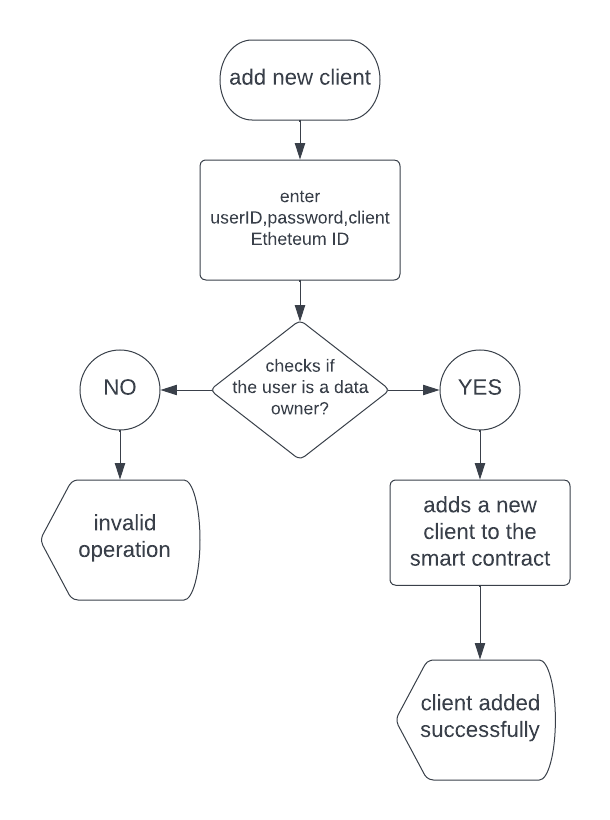


Fig 5. Adding client flowchart

*D. Adding file access to the client*

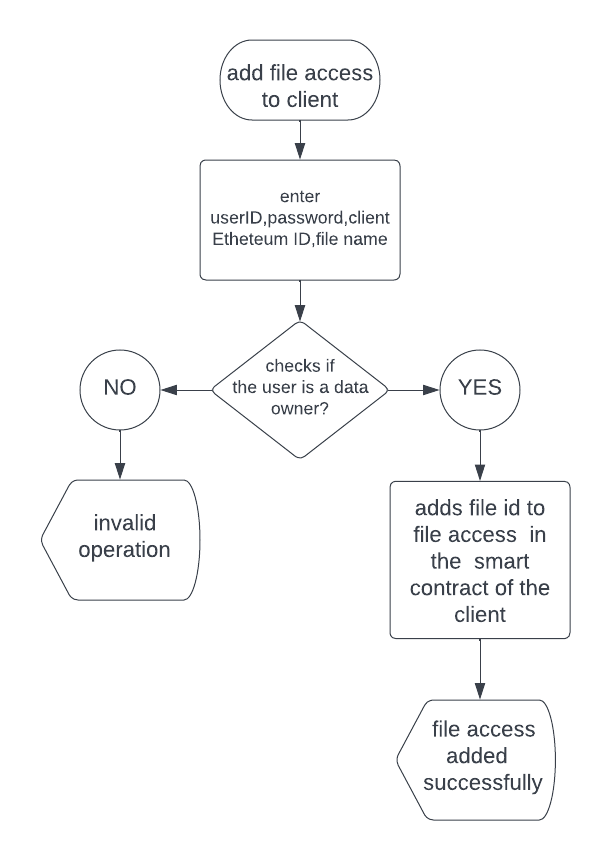


Fig 6. Adding file access to the client flowchart

When the data owner approaches the private cloud for adding file access to its client, initially the private cloud authenticates the data owner. After successful authentication of the data owner, the private cloud adds file id to file access array in the smart contract of that client.

*E. Data retrieval procedure*

In order to retrieve the data that is stored in the private cloud, the client or owner of the data will first make a request for it, and then the private cloud will carry out the process of retrieving the data.

The private cloud verifies the user's username and password to see if the file name is already stored in the database. A smart contract is used by the private cloud to check if the client has access to the requested file. The mapping table will be retrieved from the file's data owner if the client does so. Using a mapping table, the private cloud gets the chunks from the public cloud. After that, the file will go through reverse AONT transformation and GCM mode AES decryption. The mapping table copy will be destroyed after the client receives the file from the private cloud.

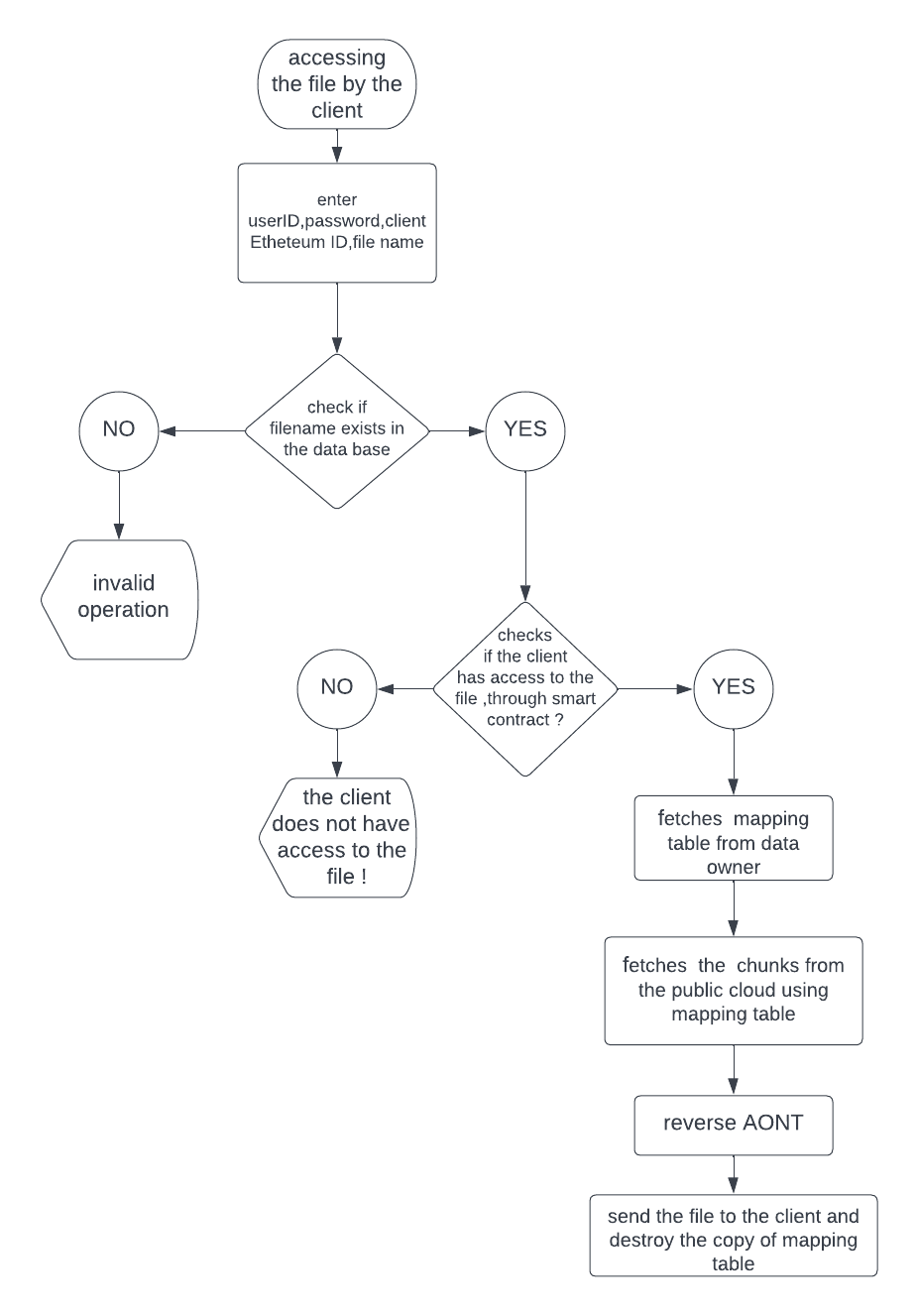


Fig 7. Data retrieval procedure flowchart

*a) Private cloud working while retrieving the data*

In the beginning private cloud collects all the N chunks of data stored in the different public clouds.Each chunk of the collected data along with (N+1) th chunk will be decrypted using symmetric key decryption method i.e, AES in GCM mode.Then N chunks will be hashed using SHA 256 standard.The N Hashed chunks are xored with their respective indices.The (N+1) th chunk is then taken and xored with all the hashed xored chunks. It results in retrieval of the AES decryption key. Then N chunks of data will be decrypted using the retrieved key.Then the data will be reconstructed using N chunks and sent to data owner/client of the data owner.

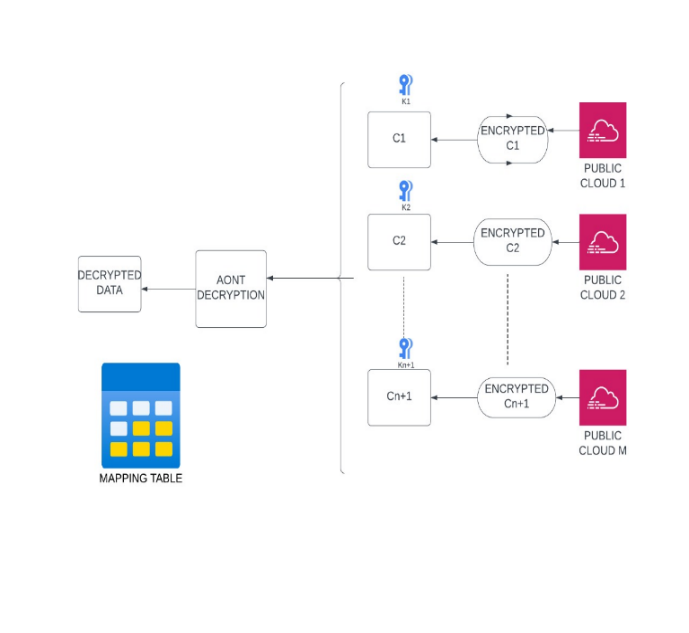


Fig 8. Private cloud working block diagram while retrieving the data

# RESULT

Fig. 9 shows the content of the inputFile.txt which is the input to the private cloud.

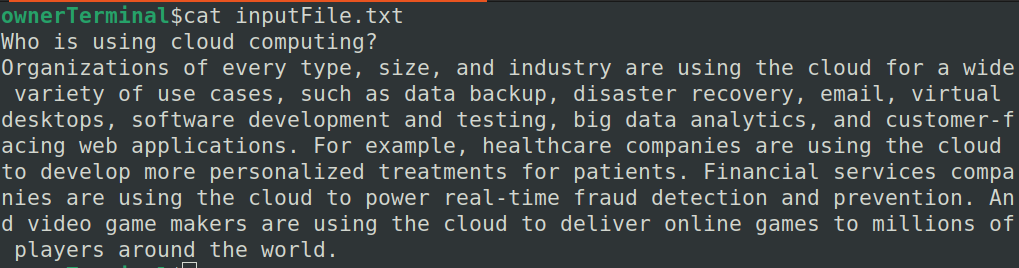


Fig 9. inputFile.txt

Fig. 10 and Fig. 11 shows the number of chunks and their details obtained after processing the inputFile.txt in the private cloud.

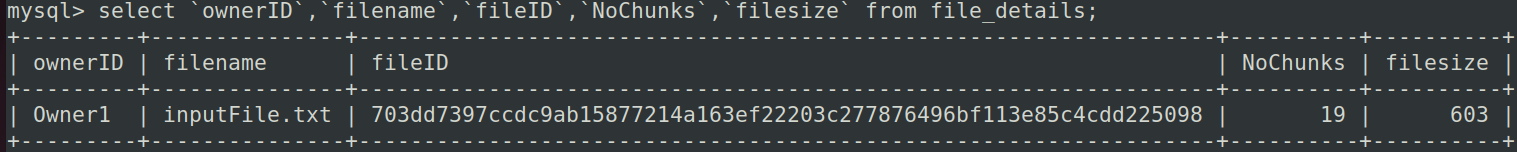


Fig 10. File details

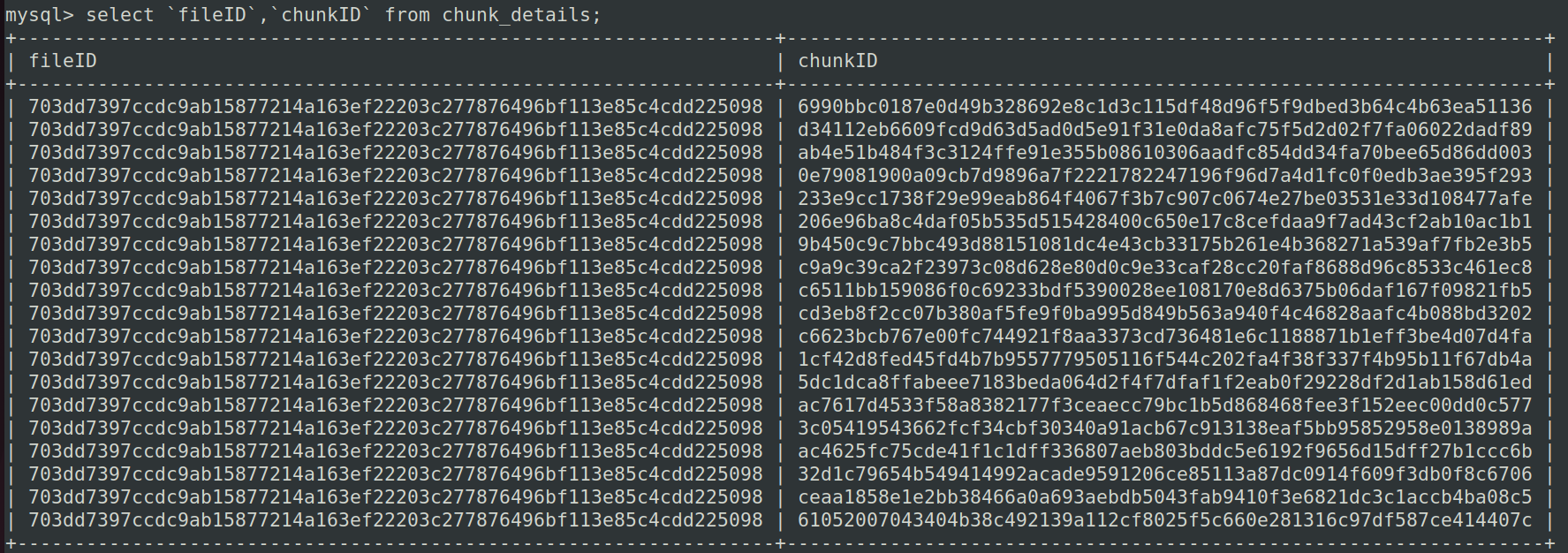


Fig 11. chunk details

Fig. 11 through Fig. 16 shows the chunks present in multiple public clouds.

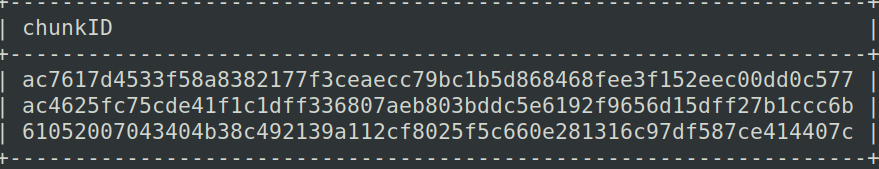


Fig 12. cloud 1

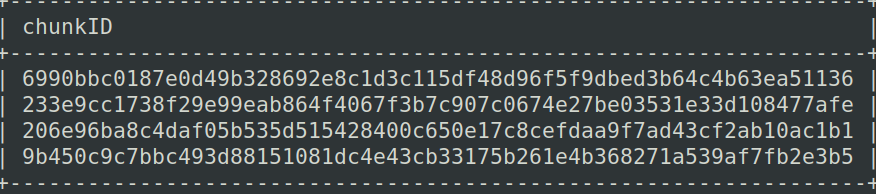


Fig 13. cloud 2

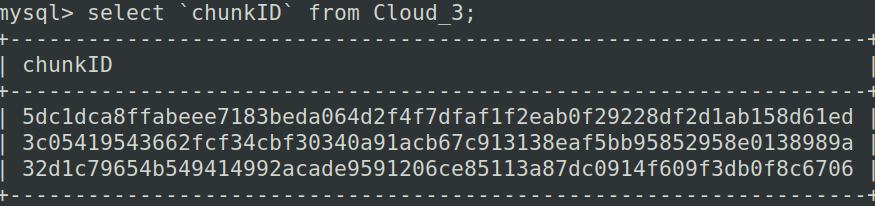


Fig 14. cloud 3

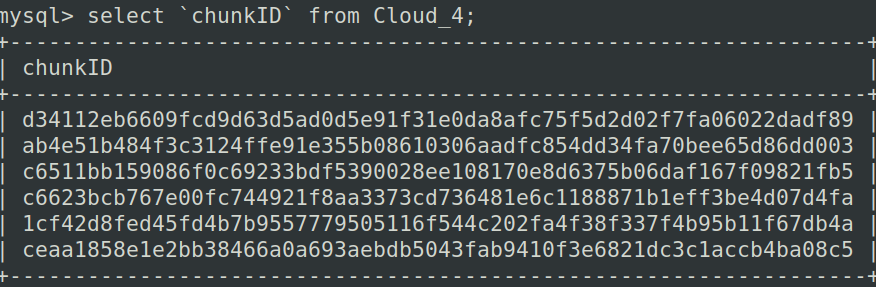


Fig 15. cloud 4

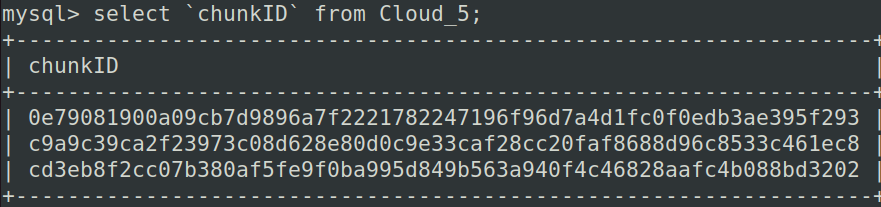


Fig 16. cloud 5

Fig. 17 shows the file that is reconstructed by retrieving all the chunks of the inputFile.txt stored in multiple public clouds and it is observed that there is no difference between the stored input file and the reconstructed output file.

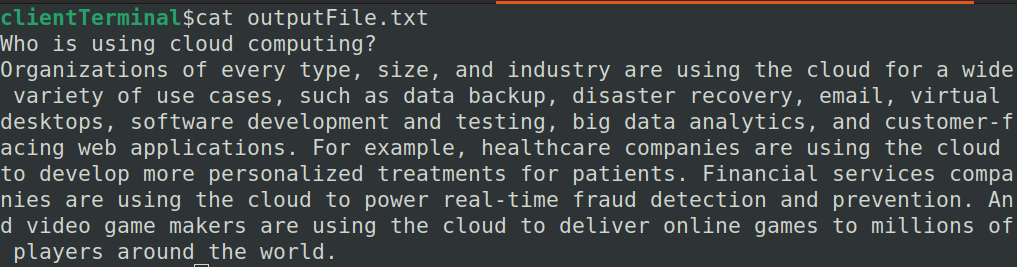


Fig 17. outputFile.txt

Data storage through this procedure significantly increase security of the data stored in the public clouds because data chunks will be distributed randomly by the private cloud. Since the mapping table is not stored in the private cloud, the private cloud also, cannot access the data without the permission of the data owner.

# CONCLUSION

In this paper, we discussed about smart contracts between the three entities the private cloud, the data owner, and their clients. One smart contract exists between the private cloud and the data owner and another smart contract exists between the established data owner and every client who wants to use this security solution/security infrastructure. We have integrated entities i.e., private cloud, distributed public cloud, smart contracts, data owner, clients using our proposed security solution/model/infrastructure, which was not used before in the present- and past-day markets as well as in businesses and by using this in premise architecture/infrastructure/solution can provide a robust and better form of security.

# References

[1] M. Shah, M. Shaikh, V. Mishra and G. Tuscano, "Decentralized Cloud Storage Using Blockchain," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), 2020, pp. 384-389, doi: 10.1109/ICOEI48184.2020.9143004.

[2] S. G. Sharma, L. Ahuja and D. P. Goyal, "Building Secure Infrastructure for Cloud Computing Using Blockchain," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), 2018, pp. 1985-1988, doi: 10.1109/ICCONS.2018.8663145.

[3] E. Bacis, S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, M. Rosa and P. Samarati, "Securing Resources in Decentralized Cloud Storage," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 286-298, 2020, doi: 10.1109/TIFS.2019.2916673.

[4] V. Amrutiya, S. Jhamb, P. Priyadarshi and A. Bhatia, "Trustless Two-Factor Authentication Using Smart Contracts in Blockchains," 2019 International Conference on Information Networking (ICOIN), 2019, pp. 66-71, doi: 10.1109/ICOIN.2019.8718198.

[5]<https://csrc.nist.rip/groups/ST/toolkit/BCM/documents/proposedmodes/gcm/gcm-spec.pdf>

[6] <https://eprint.iacr.org/2022/492.pdf>

[7] X. Shen and Z. Li, "Implementing Database Management System in P2P Networks," 2008 International Seminar on Future Information Technology and Management Engineering, 2008, pp. 528-532, doi: 10.1109/FITME.2008.76.

[8] W. Kong, Y. Lei and J. Ma, "Data Security and Privacy Information Challenges in Cloud Computing," 2016 International Conference on Intelligent Networking and Collaborative Systems (INCoS), 2016, pp. 512-514, doi: 10.1109/INCoS.2016.52.

[9] Ablahd, Ann & Dawood, S & Ablahd, Drann. (2020). Using Flask for SQLIA Detection and Protection. Tikrit Journal of Engineering Sciences. 27. 10.25130/tjes.27.2.01.

[10] H. Cheng, C. Rong, M. Qian and W. Wang, "Accountable Privacy-Preserving Mechanism for Cloud Computing Based on Identity-Based Encryption," in IEEE Access, vol. 6, pp. 37869-37882, 2018, doi: 10.1109/ACCESS.2018.2851599.

[11] S. Alatawi, A. Alhasani, S. Alfaidi, M. Albalawi and S. M. Almutairi, "A Survey on Cloud Security Issues and Solution," 2020 International Conference on Computing and Information Technology (ICCIT-1441), Tabuk, Saudi Arabia, 2020, pp. 1-5, doi: 10.1109/ICCIT-144147971.2020.9214397.

[12] M. Haifeng and Z. Ji, "Block-chain based cloud storage integrity verifycation scheme for recoverable data," 2022 7th International Conference on Intelligent Informatics and Biomedical Science (ICIIBMS), Nara, Japan, 2022, pp. 280-285, doi: 10.1109/ICIIBMS55689.2022.9971553.

[13] N. Nahar, F. Hasin and K. A. Taher, "Application of Blockchain for the Security of Decentralized Cloud Computing," 2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD), Dhaka, Bangladesh, 2021, pp. 336-340, doi: 10.1109/ICICT4SD50815.2021.9396921.

[14] R. Pise and S. Patil, "Enhancing Security of Data in Cloud Storage using Decentralized Blockchain," 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 2021, pp. 161-167, doi: 10.1109/ICICV50876.2021.9388521.

[15] S. D. Jondhale, S. K. Korde, D. S. Rakshe, I. M. Krishna, S. V. Akram and D. Verma, "Blockchain in Cloud Computing: Design Challenges," 2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India, 2022, pp. 765-773, doi: 10.1109/SMART55829.2022.10047424.