Efficient And Privacy – Preserving With Blockchain Data Secured Computing

Kaviya Devi V[1], Prashanth G[2], Thanishkaa P[3], Varnish K [4]
[1]Assistant Professor, Dept of CSE
[2, 3, 4]Dept of CSE
[1, 2, 3, 4] Sri Ramakrishna Institute of Technology

Abstract- *Before outsourcing data to the cloud for use, digital encryption is used due to concerns about cloud security. This poses a problem for effectively running queries over cipher texts. Solutions based on cryptography are currently only partially operational. Excessive computational complexity, poor generality, poor verifiability, and lack of support. In the cloud, data owners may share their outsourced data with a large number of users, who may only want to retrieve the data files that correspond to their specific searches. The most often used method for doing this is keyword-based retrieval. The proposal of a new searchable encryption scheme, in which novel technologies in cryptography community are employed, including Homomorphic Encryption. In the proposed scheme, the data owner encrypts the searchable index with Homomorphic Encryption. When the cloud server receives a query consisting of multikey words, it computes the scores from the encrypted index stored on cloud and then returns the encrypted scores of files to the data user. Next, the data user decrypts the scores and picks out the top-k highest- scoring files' identifiers to request to the cloud server. The retrieval takes a two-round communication between the cloud server and the data user.*

Keywords- Blockchain, Data secure.

I. INTRODUCTION

Cloud computing's drawbacks are compensated for by fog computing. It has numerous benefits, but there are a number of quirks that must be understood, including security, resource management, storage, and other elements simultaneously. The suggested model uses the blockchain's reward and punishment system to encourage active resource contribution from fog nodes.To create a transparent, open, and tamper free service, the fog node's actions while providing resources and the extent to which the task was completed when contributing resources are bundled into blocks and recorded in the blockchain system. It offers a five-category breakdown of threat models against IoT-based fog models, including assaults on the qualities of privacy, authentication, confidentiality, availability, and integrity. The fog computing environment can be used to deploy privacy-focused blockchain-based solutions as well as consensus algorithms for Internet of Things applications and how they will be modified for Trust Chain.Many fog nodes (FNs) are part of the Software Defined Network (SDN) core network and make up the fog computing system. In order to provide a significant quantity of computing power and storage space, fog computing refers to an architecture that places FNs and MDs at the edge. Most blockchain applications require a significant amount of computer power and storage space. We suggest a Blockchain-based fog node cluster to reduce the need for computational power and storage space (BFNC).To conserve storage capacity, a blockchain in a BFNC has a static length restriction. Moreover, BFNC, a small-scale P2P network, generates blockchains with less CPU resources than a large-scale P2P network. A group of BFNs at the edge of an SDN core network make up a BFNC (SCN). Only essential data is sent to BFNCs by a SCN's central controller. A BFNC can be thought of as a P2P network as a result.

BACKGROUND HISTORY:

In order to conduct tasks closer to end users, fog computing (FC) extends cloud computing (CC) from the centre of the internet architecture to the edge of the network. It has been demonstrated that this extension improves security and lowers latency and energy use. On the other hand, blockchain (BC) is the cryptography's fundamental technology and is used in a broad variety of applications. The research community was motivated to merge BC's distributed trust management criteria with FC in order to take a step towards developing a distributed and trusted Data, Payment, Reputation, and Identity management system because of its security and dependability. The research community was motivated to merge BC's distributed trust management criteria with FC in order to take a step towards developing a distributed and trusted Data, Payment, Reputation, and Identity management system because of its security and dependability. A new paradigm in computing and storage resource provisioning for Internet of Things (IoT) devices is called fog computing. All devices in a fog computing system can offload their data or computationally heavy operations to close-by fog nodes rather than the faroff cloud. Fog computing can dramatically reduce the transmission time between IoT devices and computer servers in comparison to cloud computing. The existing fog system is, however, quite open to malevolent intrusions.To we suggest dividing the fog system into fog node clusters (FNCs), with fog nodes (FNs) in one cluster sharing a common access control list, in order to boost security. (ACL) it has blockchain protection. Blockchain generation needs a lot of computer power and can quickly deplete FNs' computational resources. In this post, we first modify the blockchain for FNC to cut down on the amount of storage space and CPU power needed.Second, a brand-new method is created for the blockchainbased FNC (BFNC) to automatically recover ACL. Additionally, we suggest a heuristic technique to shorten the time needed to compute block hash values by working with all available devices. According to the simulation results, utilising a cooperative computing technique can speed up block hash computations compared to noncooperative ones.

PROBLEM STATEMENT:

The majority of the auditing protocols in use today are PKI-based. (Public Key Infrastructure). For PKI, managing certificates is a complicated task. PKI-based auditing protocols, however, are particularly cumbersome for batch auditing in the context of multiple users since it requires certificate verification, which might add to the auditor's workload. We suggest an ID-based public auditing approach to address this issue by fusing ID-based encryption with the homomorphic authenticator mechanism.

SCOPE OF THE PROJECT:

The main objective of this project is to develop and put into use a system that will let customers hire a third party auditor to check the veracity of their data while it is being stored in the cloud (TPA). The TPA will be responsible for checking the veracity of the users' data on their behalf. Because the TPA shouldn't have access to the substance of the user's data during the auditing process, the privacy of the user's data will be safeguarded. Also, the technology will safeguard the privacy of user data by limiting access to it while it is being stored on the cloud.

EXISTING SYSTEM:

Blockchain was used in earlier research to lower the amount of storage space and computational power needed for fog node clusters (FNC). Second, a brand-new method is created for the blockchain-based FNC (BFNC) to automatically restore the access control list. Additionally, we suggest a heuristic technique to shorten the time needed to compute block hash values by working with all available devices.Most blockchain applications, like bitcoin, require a significant amount of computer power and storage space. We suggest a Blockchain-based fog node cluster to reduce the need for computational power and storage space. (BFNC). To conserve storage capacity, a blockchain in a BFNC has a static length restriction. Additionally, because BFNC is a small-scale P2P network, creating blockchains there uses less computational power than it would in a larger P2P network.The length of the blockchain is limited to a fixed number of blocks due to the limited storage space that each BFN in a

BFNC has (previous blocks are automatically erased when the length hits the upper limitation). BFNs in a BFNC are not compensated once they have finished computing BHs, in contrast to bitcoin miners. As a result, rather than being competitive, the ties between BFNs are cooperative. That is, all BFNs in a single BFNC pool their computing resources and attempt to obtain the first BH that satisfies the system requirements as quickly as possible.

PROPOSED SYSTEM:

Utilizing homomorphic authenticators was the method. A client can outsource the authentication of a sizable group of data items, as well as the accompanying authenticators, to an untrusted server using homomorphic authenticators (HAs). Abstract Functions on authenticated data can be evaluated using homomorphic authenticators.Both in the public key setting and the secret key setting, there are constructions that take the form of homomorphic signatures and homomorphic message authentication codes (MACs). The publickey infrastructure (PKI), a key management system that supports public-key cryptography, can help with the most important need of "assurance of public key." Public key assurance is provided by PKI. It enables public key distribution and public key identification.On behalf of the users, the TPA will carry out the task of examining the accuracy of the users' data. The system will protect user data confidentiality by preventing other parties from accessing it while it is stored in the cloud. By combining a homomorphic token with distributed erasure-coded data verification, this strategy combines storage correctness insurance with data localization.Considering the time, computation resources, and even the related online burden of users, we also provide the extension of the proposed main scheme to support third-party auditing. Highly efficient and resilient to Byzantine failure, malicious data modification attack, and even server colluding atta.

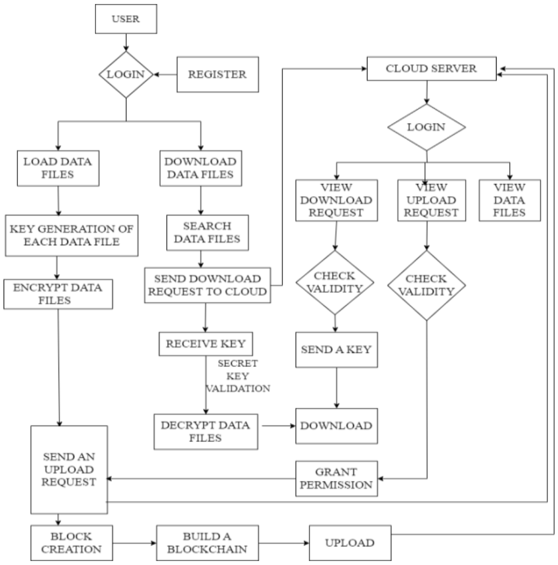


Fig.1.1. Flowchart

The flowchart describes about the cryptography in fog computing which connected to the cloud database.

II. REQUIREMENT SPECIFICATIONS:

2.1 HARDWARE COMPONENTS

- Processor : Any Processor above 2 GHz
- Ram :1 GB
- Hard Disk : 80 G

2.2 SOFTWARE COMPONENTS

- Language : Java, J2EE
- Technology : JSP, Servlet
- Database : MySQL 5.0
- Backend Tool : SQL Yog
- Developing Tool : NetBeans IDE 7.2.1
- Web Server : Apache Tomcat 5.5
- Build Tool : Apache Ant.

2.3 SYSTEM SPECIFICATIONS:

2.3.1 Symmetric-key algorithms

The term "symmetric-key algorithms" refers to cryptographic algorithms that encrypt plaintext and decrypt ciphertext using the same cryptographic keys. There can be a straightforward transformation that connects the two keys, or the keys might be identical. In reality, the keys stand for a shared secret that two or more parties might use to keep a link to confidential information open.One of the main disadvantages of symmetric-key encryption in comparison to public-key encryption is that both parties must have access to the secret key. (also known as asymmetric-key encryption). But for bulk encryption, symmetric-key techniques are typically preferable. They feature a lower key size, which translates to less storage space and speedier transmission, with the exception of the one-time pad. As a result, symmetric-key encryption frequently uses asymmetric-key encryption to replace the secret key.

2.3.2 KEY GENERATION

In symmetric-key algorithms, both the sender and the recipient of a message must be in possession of the same secret key. Any early cryptographic system cannot function without sending a copy of the secret key to the sender or recipient across a physically secure channel. The majority of contemporary cryptographic systems still internally encrypt messages using symmetric key algorithms, but they do away with the need for a physically secure channel by securely deciding on a new secret key for each session or conversation using Diffie-Hellman key exchange or another public-key protocol. (Footnote secrecy).

2.3.3 HOMOMORPHIC ENCRYPTION

Calculations can be performed directly on encrypted data without having to first decrypt it thanks to a type of encryption called homomorphic encryption. When decrypted, the calculations' results are identical to what they would have been if they had been performed on the unencrypted data. The calculations are then saved in an encrypted format. Using homomorphic encryption, compute and storage can be outsourced discreetly. This makes it possible to encrypt data while it is processed in a commercial cloud environment.A type of encryption known as homomorphic encryption enables users to do additional analyses on encrypted material without needing to know the secret key. The outcome of such a calculation is encrypted. Homomorphic encryption can be seen as progression of public-key cryptography[how?]. It is possible to consider the plaintext and ciphertext spaces as homomorphisms between the encryption and decryption operations. The word "homomorphic" refers to the algebraic concept of homomorphism.

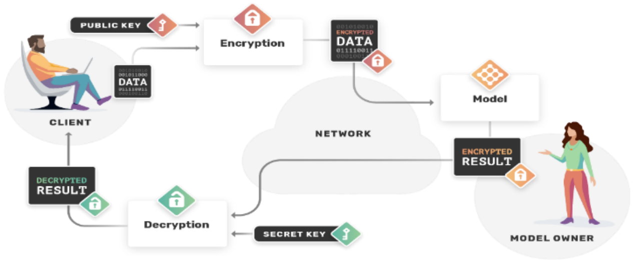
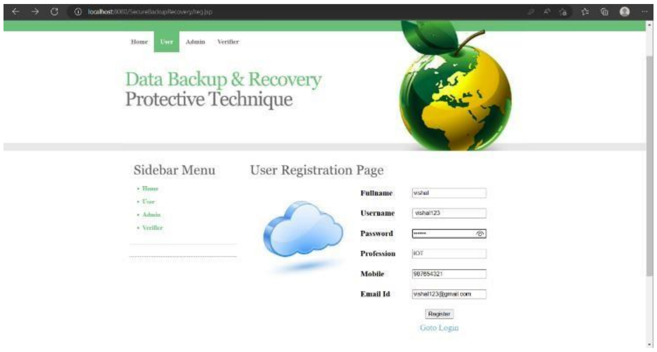


Fig.1.2. HOMOMORPHIC ENCRYPTION

This figure shows the homomorphic encryption in the cryptograph.

2.3.4 OUTPUT SCREEN

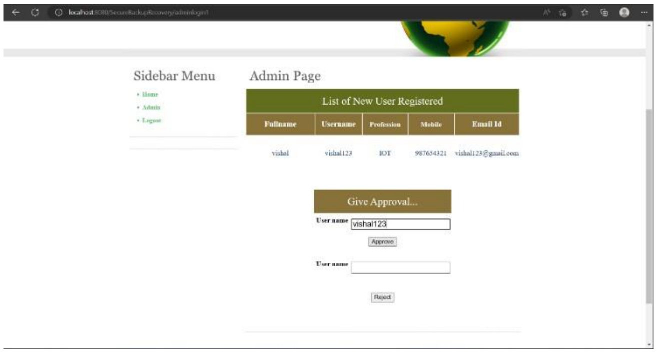
USER REGISTRATION



User Registration Page

Fullname: vishal
Username: vishal123
Password: ••••••
Profession: IOT
Mobile: 987654321
Email Id: vishal123@gmail.com

ADMIN PAGE



Admin Page

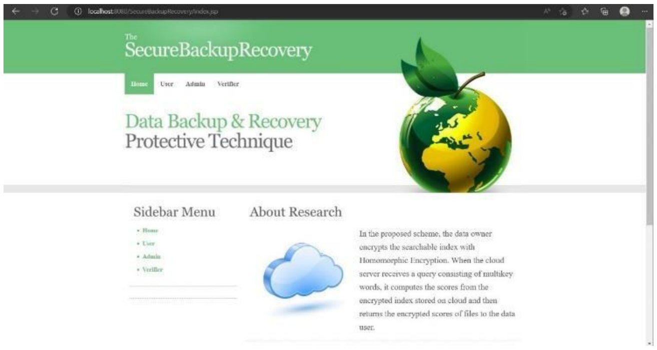
List of New User Registered

Fullname	Username	Profession	Mobile	Email Id
vishal	vishal123	IOT	987654321	vishal123@gmail.com

Give Approval...

User name: vishal123

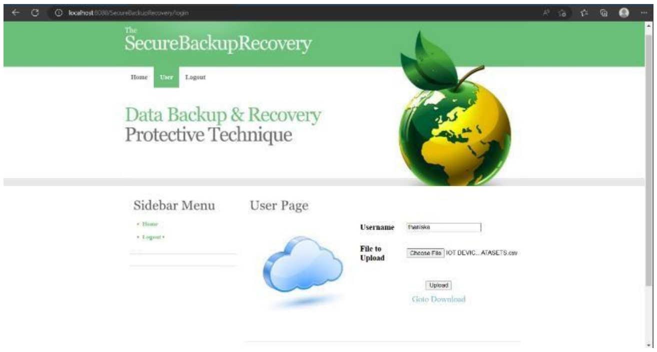
User name:



About Research

In the proposed scheme, the data owner encrypts the searchable index with Homomorphic Encryption. When the cloud server receives a query consisting of multikey words, it computes the scores from the encrypted index stored on cloud and then returns the encrypted scores of files to the data user.

UPLOAD PAGE



User Page

Username: thaniska

File to Upload: IOT DEVIC...ATASETS.csv

III. CONCLUSION

3.1 SUMMARY:

In this method, the issue of data security in cloud data storage, which is essentially a distributed storage system, has been researched and examined. We provide an efficient and adaptable distributed architecture with explicit dynamic data support, including block update, delete, and append, in order to ensure the integrity and availability of cloud data and enforce the quality of dependable cloud storage service for customers.In the file distribution preparation, we rely on erasure-correcting code to deliver redundancy parity vectors and ensure the dependability of the data. The integration of storage correctness insurance and data error localisation is accomplished by this technique by combining the homomorphic token with distributed verification of erasure-coded data.

3.2 FUTURE WORKS:

We also offer the extension of the suggested main scheme to facilitate third-party auditing, taking into account the time, computation resources, and even the associated online load of users. We demonstrate that this approach is highly effective and resistant to Byzantine failure, malicious data modification assault, and even server collusion attack through rigorous security and extensive experiment findings.

REFERENCES

[1] A Cooperative Computing Strategy for Blockchain-secured Fog Computing, by Di Wu and Nirwan Ansari, IEEE Internet of Things Journal, Volume: 7, Issue 7, Pages: 6603–6609, July 2020.

[2] Enabling Identity-Based Integrity Auditing and Data Sharing With Sensitive Information Hiding for Secure Cloud Storage, IEEE Transaction on Information Forensics and Security, W.Shen, J.Qin, J.Yu, R.Hao, and J.Hu, (Volume: 14, Issue: 2, Page(s): 331-346), Feb 2019.

[3] Decentralized and Privacy-Preserving Public Auditing for Cloud Storage Based on Blockchain, (Volume: 8, Page(s):139813-139826), Y. Miao, Q. Huang, M. Xiao, and H. Li, IEEE Access, July 2020.

[4] IEEE Transactions on Wireless Communications, "Control-data separation with decentralised edge control in fogassisted uplink communications," Volume: 17, Issue: 6, Page(s): 3686–3696, June 2018. J. Kang, O. Simeone, J. Kang, and S. ShamaiShitz.

[5] Blockchain meets IoT: an architecture for scalable access control in IoT, O. Novo, IEEE Internet of Things Journal, Volume: 6 Page(s): 99, 2018.

[6] Practical Homomorphic Message Authenticators for Arithmetic Circuits, Catalano and Fiore, Journal of Cryptology, volume 3, page 37, 2018.

[7] "Homomorphic Encryption Methods Review," 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, N. N. Kucherov, M. A. Deryabin, and M. G. Babenko.