Journal of Engineering Science and Technology Review 16 (2) (2023) 123 - 130

Research Article

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

Block Chain Based Secure with Improvised Bloom Filter over a Decentralized Access Control Network on a Cloud Platform

Ravikumar Ch*, Isha Batra and Arun Malik

Department of Computer Science & Engineering, Lovely Professional University, Punjab, India-144411,

Received 11 November 2022; Accepted 27 March 2023

Abstract

The control of access to data is perhaps the primary factor in enhancing secure data within this cloud-based storage system. But, traditional data sharing, along with access control techniques, have been a significant problem in the field of research that includes leakage of privacy information and key misuse. Transferring Electronic Records (ER) on the cloud has allowed data sharing among various health applications. Therefore, two major contributions are created in this study to provide effective access control and private data sharing as well as retrieval within the cloud based on blockchain. In the initial contribution is blockchain-based access control and sharing method effectively resolves the single point of failure within the cloud infrastructure. The Data User (DU) generates an application for registration based on the ID and password and is then forwarded to the Data Owner (DO). The DO data is incorporated into a transactional Blockchain based by encrypted master keys. The DO secures data while the encrypted file is uploaded into the Interplanetary File System (IPFS). In the second piece of work blockchain structure was created to facilitate information sharing as well as retrieval within cloud platforms.

Keywords: Cloud computing, decentralized access control, Blockchain technique, Data Integrity.

1. Introduction

Cloud computing can be described as a brand new technology that has lately sparked an interest from both academia and business. With cloud computing, users can use a variety of internet-connected services provided by software without the need to buy and install these on their personal computers with an internet connection. Based on the National Institute of Standards and Technology cloud computing (NIST) the access to a database of computer resources that can be changed can be a model for providing valuable, timely information. Today the management of huge volumes of data for massive international companies has proved difficult. Since cloud storage has greater archiving, distribution and uploading capabilities, many companies have shifted to cloud storage. Data confidentiality and integrity and helping to protect data are the primary concerns cloud computing has to deal with.

The majority of people choose to store their personal data on the cloud. There are some security and copyright concerns however. The fact that information can be accessed by anyone outside the person who owns it is the primary problem with data being sent to an outside environment. Cloud service providers do not offer the level of privacy and security necessary for effective data security and security. The advent of a decentralized cloud-based storage network provides a number of advantages over traditional data center storage. Similar to traditional storage methods decentralized cloud storage networks provide data security using encryption that is client-side. However, managing encrypted data is not without numerous issues, with usability being the most crucial. In particular, the owner must be able to authorize others to gain access to remote encrypted data and obtain useful material that is not finished. The entire set of data, then filtering the data, then giving the appropriate portions to the authorized client is a simple approach. But the high cost for the consumer makes the option ineffective and defeats the purpose outsourcing data.

1.1. Access Control-based Data Privacy

Security and privacy are the most important requirement in cloud computing, and it opens the way for the development of the most secure and efficient method of controlling access to information sources. This has become an essential issue for cloud computing4. Access control methods limit the possibility and probability of the requester accessing information following the confirmation of the authentic identity of the user. But, the access control method is usually employed to safeguard the important data resources, and to prevent the illegal entry of intrusions. The various access control techniques that are available include the discretionary access control (DAC), Role based access control (RBAC) as well as utilization control (UCON) as well as attribute-based access control (ABAC) and as well as mandatory access control (MAC). As compared to the many conventional approaches the access control model is more prominent in cloud-based frameworks. When using cloud computing storage services and storage cloud computing, users must allow authentication to the cloud provider (CSP) and also manages the proper guidelines for authorizing information and services. To confirm the security of cloud computing it is vital to ensure access control between the service providers and mutual authentication. Additionally, cloud users don't solely can control the channel attack, but also have other ways to ensure the security of information. In recent years, CSP exploits a variety of access control

*E-mail address: chrk5814@gmail.com
ISSN: 1791-2377
doi:10.25103/jestr.162.16

mechanisms within the cloud domain to provide more security but there are some limitations in place.

1.2. Blockchain-based Access Control

The model in [5], demonstrates the decentralized privacy solution that is designed to protect the confidentiality of information that is collected and managed through an external third party. The method is built on blockchain technology that functions as an access controller to guarantee the transparency of the privacy information that contains points and the off-blockchain Distributed Hash table (DHT) that needs to be authorized with blockchain in a manner that encrypted data can be kept. When a user logs in to access their account, the latest compound identity is constructed and shared. The identity is comprised of login key pairs for both the service and user and an identification key used to decrypt and encrypt the data. The blockchain is used to verify the identity of the user. It also checks whether the service is authorized to access the data. It creates the hash code to retrieve the data from off-chain storage.

Bloom filters are defined by a target false positive rate. By setting the desired false positive rate in Bloom filtering, Bitcoin designers aim to give a decent anonymity level to conceal the addresses. The Simplified payment Verification (SPV) light flowering network [13], was expanded to utilize Bloom filters to process transactions involving nodes. Due to the constant growth of blockchain-related applications, SPV nodes have become the most widely used type of blockchain nodes.

Data security is the top issue for those who are interested in cloud computing. To reduce user issues, cloud computing must be supported with the best security methods and principles. The majority of users of cloud services are worried about the possibility that their personal data will be shared with other cloud service providers, or used to serve purposes that are not related. Blockchains keep the record of all data transferred during transactions, and it is virtually impossible to alter the information once it's been entered. In comparison with other methods of security it is much easier to utilize and more effective.

To solve this problem to address this issue, this study presents an approach that utilizes Blockchain's Secure Data Storage and Access System to store data and access control through a bloom-counting filters. This is why we recommend that you use Blockchain as a trusted technology for the development of smart contracts which use computer protocols to automotive jobs and reduce the amount of time required to complete various business procedures. Blockchain is a peer-topper network that keeps track of transactions in a decentralized and secure electronic ledger. The ledger, shared by all participants in the network is able to keep records of all transactions among nodes of an organized chain of digital hash-linked cryptographic block.

2. Related Work

2.1. Ledger-based Blockchain Model

The current research works that are that are based on blockchain-based ledger techniques can be explained as follows: Rajput, in accordance with [12] has developed an Emergency Access Control Management Model (EACMS) to manage health records that utilizes blockchain technology. This model incorporated a variety of rules were applied using smart contracts for controlling emergencies and for managing time duration. In addition hyperledger composer was utilized to create Business Network Archive (BNA) that identifies the system's capacity. Every transaction was impacted by data retrieval and authorization from ledgers, which were performed via smart contract. In addition, this model is dependent on smart contracts in ledger to provide a reliable security, auditable and secure system.

In [13] the company introduced a blockchain-driven access control model with numerous attribute authorities in secure cloud-based data sharing model. The model employs Shamir secrets sharing as well as Hyper ledger Fabric technique was designed to implement the process, in which each attribute was managed with various authorities in order to avoid one-point failure. Additionally, the benefits of the blockchain model were evaluated to establish trust between various and create smart contracts for the estimation of attributes with tokens that reduce the processing and communication burden for the data user. In addition, the blockchain model is helpful to record access control methods with a secured and auditable method.

In [10], the blockchain was developed as a decentralized model for data storage. File metadata were stored on blockchains, while actual files were collected through Distributed Hash Tables (DHT) in various locations built on the peer to peer network. In addition, the invented off chain storage model achieves very low latency as well as high throughput. This method reduces the project's dependence of central resources to process as well as uptime and accumulation.

2.2. Encryption-based techniques

This section provides information about the research conducted on various methods of encryption used to access control and sharing data process.

In [14], the Blockchain-based model was developed which is distributed Key Management Architecture (BDKMA) to control access within the Internet of Things model. This model uses the fog computing model was employed to decrease delay and the multi-block chains were run in cloud computing to obtain cross domain access models. Blockchain model was utilized to satisfy the requirements of extensibility, fine grained auditability, decentralization and high capacity and privacy-preserving requirements for control of access. In addition, systems operation models along with various authorization assignment nodes as well as group access models were developed to facilitate the system's extensibility. The dynamic duration of transaction collection changes allow the system to be able to cater for a variety of types of structures. This technique improves the efficiency of the system as well as scaling capacity in relation to the size of network.

In [15], we designed a blockchain-based security system that uses secure mutual authentication. It is called BSeIn to enforce precise system of access controls. This method typically involves multi receiver's encryption and the integrated attribute signature as well as message authentication code. In addition, it offers security and security assurance. In this case, blockchain and attribute signatures were used to authenticate terminals authenticate gateways, and message authentication codes. Additionally, the multi receiver encryption was utilized to ensure confidentiality. Smart contracts were used to ensure that the process was scalable and was made possible by the interplay with smart contracts.

In accordance with [16] the modeled cross domain sharing method for electronic health information. In this approach, cryptographic models were used to making it

possible to share data in a secure manner for privacy protection of patient data. This technique effectively incorporates Revocation on demand and fine-grained access control. Additionally, access control was implemented by using a delegation model. Furthermore, the delegation process was incorporated by proxy signature-based model and role-based model.

2.3. Smart Contract-based Blockchain Methods

The current literature review of research studies that are based on blockchain-based Smart Contract methods is explained as follows,

In [17] we created Blockchain-based access control and permission delegation for IoT (BACI). This model was created to allow delegation, and the control of access for IoT model. It is extremely dependent on event and query base permissions allocation. Following that, the blockchain structure was used to provide decentralized, delegation services, which are reliable secure and verified structure. The node with the least permission group was assigned along with the essential permissions were delegated to each nodes in the event. The model is comprised of a variety of components, including the device used by the user, IoT manager, IoT device, blockchain manager application manager, smart contract and blockchain. This model is efficient in controlling access control of restricted devices.

In [18], the presented the data sharing model, along with access control method that relies on smart contracts based on blockchain in IoT. This method was designed to address a variety of issues related to the authentication process and trust in security of access within IoT. In this case, multiple smart contracts, such as Register Contract (RC), Access Control Contract (ACC) and Judge Contract (JC) were used to provide effective access control management. ACC is the one that controls access systems and RC is used to authenticate users within the systems. Furthermore, JC applied behavior judging model to detect the person who was committing the offence. Once the misbehavior identification was complete the penalty was then determined for each subject.

3. Proposed Methodology

Figure 1 shows a clear and concise description of this strategy. This framework was created in [11, 12, 13] as a result of research inspiration. Within this paradigm, data is shared between two sets of users, or peers, known as subject and object. The primary focus is on a consumer who genuinely wants to work with service providers. That is, the purpose is to be the source of knowledge that the consumer needs. Resources provide information, files, programmers, and so on. Furthermore, triple contracts are used to manage data sharing and access between objects and subjects.

3.1. Smart Contracts

The conceptual technique is made up of three contracts (ACC, RC, and JC) that are used to manage and provide network users with multiple security alternatives. ACC is in charge of overseeing access control across the entire framework. The registry contract grants authorization and authentication for the subject or object, as well as the maintenance of all entries in the database storage. The Judgments Contract monitors the subject's or object's negligence (JC). If misconduct is discovered during the trial, the JC is a legal requirement for the individual accountable.

3.2. Access Control Contract

It is the primary intelligent contract that ensures IoT node privileged access. When it is necessary to access any facility from the object, it processes the request in an acceptable manner. The ACC then saves the recipient's data access. This improves the service's effectiveness. This is why, in terms of cost and time, technology is evolving. Furthermore, device computation time is usually executed. When a user sends inquiries to the device, the ACC is employed.

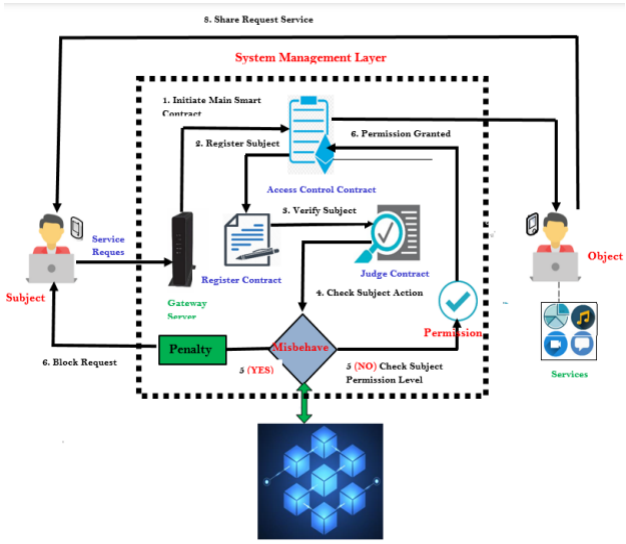

Fig. 1. The Proposed Data Sharing and Access Control Model.

3.3. Register Contract

Maintain a registry that maps a smart contract's name and the address of its most recent version as a (name, address) pair. Look up the registry to get the address of a smart contract before activating it. Blockchain-based applications, like any other software application, must be upgraded to new versions.

3.4. General Working Procedure for Secure Data Sharing

To carry out the data sharing process within cloud platforms, it's necessary that only authorized users have access to the cloud data. If the person who owns the data has to provide their data with an individual group, the owner of the data sends the key to secure the data to every group members. In addition, any members of the group could get the encrypted data from the cloud and the decryption of the data happens by using the key. Therefore, the member of the group is not dependent on the intervention by the owner of the data. The protected data is transferred using transforming the original data on healthcare into protected from privacy to aid in the safe transmission process. Additionally, the method employed privacy and utility parameters to control the crucial data. When a request is made for encrypted data stored in the cloud platform and the CSP examines whether the requested data stored in storage or is compatible by the terms of the request with the help of a data retrieval system. When data retrieval is performed, initial data are extracted from encrypted data.

4. Access Control and Data Sharing Model in Cloud Decentralized Storage with Block Chain System Using Improvised Bloom Filter

The access control and sharing method is developed for cloud storage that is decentralized. The model of decentralized storage that has been developed typically comprises eight steps, including setting up, encryption, user identification, generation of tokens control testing, setup and decryption. Additionally, the decentralized storage is comprised of four entities, including DO, Smart agreement, DU as well as Transactional blockchain [3]. Each entity performs its role to manage access control and data sharing models. The most common scenario is that a DO is a collective or individual and the DO is the owner of the file in the sharing. In contrast, DU is a DO client for data, who have the right to observe files. DO set up the stage-through encryption of master key, and then integrating the master key into the transactional blockchain. Smart contracts are placed in a transactional blockchain that is that is based on DO. But, the smart contract is utilized to keep track of the encrypted keywords and provides a search engine that is effective to DU. In the setup phase, DU transfers the registration request to DO. In addition, SO encrypt and uploads the file to IPFS and embeds the metadata for ciphertext into the transactions blockchain. Furthermore, DU downloads file from IPFS and then decrypts it.

Bloom filters use what is called an input data structure array. The array is of a length or capacity of storage sufficient to meet the requirements. It means, when you build bloom filters, you can decide how long the filter's length is, in accordance with the needs. The number of entries to add to the basic data structure, as well as the number of i^{sh} functions will be utilized within the filter, combining all of these inputs.

Additionally in the process of the design, it should be considered that hash functions have to begin at 0 and finish at the entries currently in use plus 1. For instance that if a flower filter is intended for 10-entry entries, the bloom filter will begin at number 0 and finish at number 9. If the design is for twenty entries, then the filter will begin at zero and end at the number 19. A design method which seeks to maximize the use of the efficiency of processing filters.

If the existing entries show every value at 0 it implies that the data does not belong within the bloom filters. Therefore, it is not in the bloom filter. When you begin adding data or other elements into the bloom filter that information will be processed by the appropriate hash functions which will then place the data at the appropriate location inside the bloom filters. This means that these places will be reflected as 1 indicating that they have elements that have already been studied.

4.1. Hash Functions Inside Bloom Filters

When setting up a bloom filter when setting up a bloom filter, completely independent and evenly distributed hash functions need to be employed. These functions enable an identifier to be assigned any kind of data which is then used to compare or index the data in the set of data.

When we speak of hash functions, we are talking about the most well-known SHA-256, MD5 or other functions such as CRC32. However, with bloom filters, it is important be aware. Utilizing a variety of hash functions increases security, but it also creates more complexity and time-consuming. Therefore, it is important to select the right functions to ensure that their capabilities are fully utilized.
However, the unidirectional nature of hash functions means that an identifier can be identified or constructed by a data element or element however the reverse process is not possible. If a user finds an identifier, they'll not be able to determine the elements or data associated with the identifier are.

Some thin clients in the Blockchain network use the Simplified Payment Verification (SPV) technique to validate transactions without needing to maintain a complete duplicate of the blockchain. As a result, only the deals in which they are interested in getting changes are specified using bloom filters by these thin nodes.

Computers that execute the Blockchain core software are known as nodes on the Blockchain network. They are referred to as full nodes if they have a complete duplicate of the Blockchain network; otherwise, they are thin nodes. You must be operating a complete server in order to participate in mining (and compete to organize transactions into blocks, solve cryptography problems, and ideally make some freshly generated Blockchains). While many full nodes on the network do not mine, they do add to the network's security and decentralization by merely verifying transactions and maintaining a complete record of the blockchain.

Thin peers, on the other hand, keep a fragmented duplicate of the database. There are a variety of causes for this, but the one we'll talk about today is for Simplified Payment Verification. A catalogue of events for which the thin node is interested in getting changes will therefore be disseminated out by the thin node. The network's complete nodes will then only provide the thin nodes with updates for the events that fit the group for which they have asked. As a result, the thin client can validate transaction data without needing a complete duplicate of the database.

It is essential to note that SPV nodes may not show every transaction in which they are potentially interested because there may be thousands of them. Instead, they compute a bloom filter for the relevant set of transactions and disseminate it to all nodes.

The Hashing Operation, which is crucial to this probabilistic data structure that we will further explore, and the Bloom Filter Data Structure are closely related. This data structure has the benefits of being quick and space-efficient while having the drawback of being probabilistic in nature.

Bloom filters are highly efficient, but their fundamental flaw is that they are probabilistic. You can comprehend this better if you use a simple example. A new element that has to be located in a list may occasionally not be totally there, but there is also a chance that it will be in the collection. If there is a chance that the element is in the collection, the result may be either a true positive or a false positive. The element can be indicated as a true negative if it is not totally present in the collection. Because Bloom Filter is probabilistic, it occasionally produces erroneously positive findings, which means that although the element is supposed to be present, it is actually absent.A Bloom Filter never indicates a misleading negative outcome, meaning it never indicates that an element is absent when it is truly present.

Before a component enters the Bloom Filter, it is subjected to k hash operations. We first establish the "k" hash function and an empty bit array with all of the elements set to zero. The number of items to be hashed and the size of the database at our disposal determine the appropriate number of various hash algorithms to use.

All k hash algorithms are used to hash the required information, and in each instance, the bit in the hashed location is set to 1. We must hash the search query once again and check to see if the bits are present or not in order to determine whether an element is already included in the Bloom Filter. The Bloom Filter may yield some accurate but

misleading results when the element is genuinely absent but is claimed to be present. The formula to determine the likelihood of receiving a false-positive result in a k-bit matrix is 1-(1/k).

The pseudo code for insertion of an element in the Bloom Filter on blockchain is as follows:

```
function insert (element){
    hash1=hashfunction (element)% Size_Of_Array
    hash2=hashfunction2 (element)%Size_Of_Array
array [hash1]=1;
    array[hash2]=1;
}
```

The pseudo code for searching an element in the Bloom Filter is as follows:

```
function search(element){
      hash1=h1(element)%Size_Of_Array;
      hash2=h2(element)%Size_Of_Array;
      if(array[hash1]==0 or array[hash2]==0){
            return false;
      }
      else{
prob = (1.0 - ((1.0 - 1.0/Size_Of_Array)**(k*Query Size)))
** k
      return "True";
```

The Blockchain network's lightweight servers are unable to synchronies the complete Blockchain's data. They interact with complete nodes, which are the holders of the entire blockchain, using the Simplified Payment Verification (SPV) algorithm in order to obtain information that is pertinent to them. The SPV system employs bloom filters to address recognized privacy concerns. However, the utility of Bloom filters is still in doubt. In this study, we examine how Bloom filter proposals behave in relation to their desired False Positive Rates (FPR). Our results demonstrate that tiny FPR values have very little bandwidth needs, whereas larger values are not appropriate for use with bandwidth-constrained devices. At the same time, we affirm that Bloom filters' privacy consequences are still relevant in the present, even though more and more new addresses are being added to the Blockchain network on a daily basis. We also investigate the impact of numerous parallel queries from lightweight clients with Bloom filter support on the CPU, RAM, and disc utilization of complete node clients.
Let's say we have 10 buckets, and three hash functions. So, when information goes through the bloom filter, it will return three identifiers within the range 1-10. This is clearly a simplified model as we can have a number of buckets as well as m of have **h** function.

In the case below, we're trying to figure out which are the cryptocurrencies Alice.

First, we run Bitcoin (BTC) through the bloom filters, and it provides the following identification numbers:

1				5		7			

Next, we put Ether (ETH) through and see the following identifiers:

	2			5				9	

Finally, we put Litecoin (LTC) through and receive the following identifiers:

		3	4						10

The bloom filter is therefore complete with:

1	2	3	4	5		7		9	10

Then, we can inquire from the bloom filter if Alice is using Ether (ETH) or not, and as the identifiers of Ether (ETH) include 2, 5 and 9. We can simply verify that one of these buckets in the bloom filter is fully filled. It is crucial to keep in mind the following point: any data that is processed by a hash function multiple times will produce exactly the same result. Therefore, every time that 'bitcoin' is processed through the hash function, the same identifiers will appear. So, if buckets 2 and 9, are all complete, so we can conclude that Alice is a user of Ether.

We can then check the bloom filter to see if Alice is a member of Ether Classic (ETC) which includes identifiers 6, 8, 9. However, even though bucket 9 is fully filled however buckets 6, and 8, aren't so we can conclude that Alice is not the owner of any Ether Classic (ETC). But if we ask if Alice owns any Monero (XMR) that has the identifiers 1, 4, and 10, we will receive an untrue positive as these buckets were filled with the mixture of Bitcoin (BTC) along with Litecoin (LTC). Therefore, it is important to recognize that although bloom filters don't reduce the chance of false positives, it can nevertheless reduce the risk of dangerous false negatives.

To reduce the possibility of false positives, a bigger bloom filter that has many buckets is a good idea.

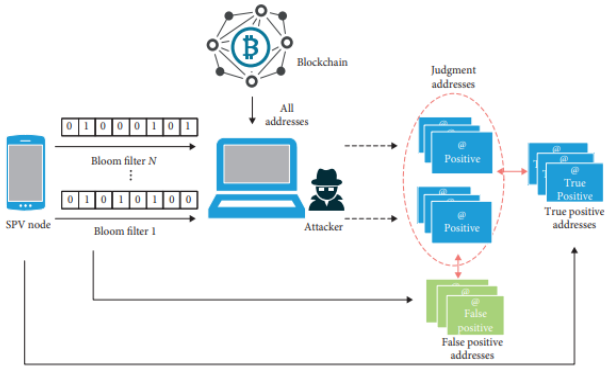

Fig. 2. SPV node attack model based on Bloom filter.

SPV node and Bloom filter are the two sides of the game. Their State set can be described as a finite one therefore it's an extremely strategic game.

This earns can be viewed as a benefit the SPV node receives from permitting the addition of address numbers to Bloom filter Bloom filter while in a secure state. If we assume that each time an address is entered into the Bloom filter then the SPV node's income is s_earning_State1 and every time an address is added.

5. Experimental Setup & Results

The implementation of the access control and privacy-based blockchain-based retrieval and sharing of data model is executed using PYTHON tool, which comes with Windows 10 OS, 4GB RAM as well as the Intel i3 processor.

A comparative study of the developed blockchain-based access control as well as privacy-based methods for sharing

and retrieving data are discussed in this section. Comparative analysis for access control techniques based on blockchain using various sizes of blockchain like 100, 200, 300, 400, and 500 is described in this section.

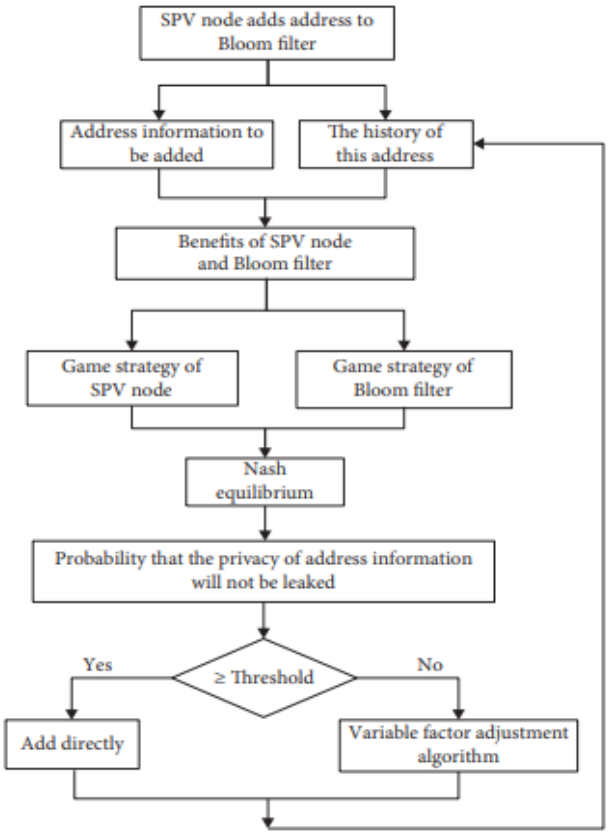

Fig. 3. Privacy protection method.

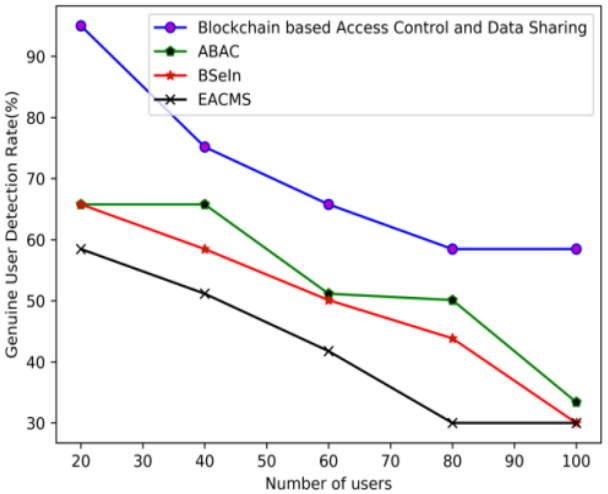

Fig. 4. Comparative analysis using genuine user detection rate with blockchain size 100.

Figure 4 provides the comparative analysis of real users' detection rates for a variety of numbers of users. The actual detection rate in ABAC, EACMS, BSeIn and other blockchain-based access control and sharing of data is 58.46 percent, 65.76%, 65.76 percent and 95% for a 20 many users. Additionally, the actual detection rate achieved through ABAC is 51.15 percent, BSeIn is 58.46%, EACMS is 65.76% and the blockchain-based access control developed and data sharing rate is 75.16 percent, while the number of people using it is. If the number of users is 60, real detection rate for ABAC, BSeIn, EACMS and Blockchain-based access control as well as data sharing are 41.75 percent, 65.76%, 51.15 percent 51.15%, and 50.10 percent. The real rate of detection in ABAC is 30 percent, BSeIn is 43.84% EACMS has 50.10 percent, and the developed method is 58.46 percent, and then there are 80 users. The real detection rate for ABAC, EACMS, and BSeIn is 33.40 percent, 30 percent 30 and 33%, respectively, while the developed model is 58.46 percent per 100 of users.

This section explains how proposed experimentation is conducted by using Ethereum technology. Ethereum is an open blockchain that defines a peer-to-peer system which is secure and safe for managing and demonstrating applications code, referred to as smart contracts. Furthermore, Python programming is used to write code using Colab notebooks. It allows anyone to write and execute arbitrary Python code by using the browser. To test the performance of the algorithm used presented in this article, analyzing the proposed method results using three methods, including memory sizes which is 180823 bits or 83677 bits. Bloom Filter is an algorithm for probabilistic data structures that is utilized to verify whether an element is part of the set or not. Bloom Filter is considered to be efficient in space since it only uses bits for the storage of information.

Here the proposed work False Positive Rate is calculated by using below equation,

$$\frac{FN}{FN + TP}$$

Where, FN is the No. of false negatives and TP is the No. of true positives (FN+TP being the total number of positives).

5.1. Using Memory Size of 180823 Bits in Improvised Bloom Filter

Figure.5 shows the practical implementation and FPR observations when taking 180823 bits size. In the graph, the X-axis indicates the unique elements fed and unique elements tested, and Y-axis indicates the false positive rates (FPR). The FPR initialized as 5.0 %. And the Practical observation average is 4.302708654133927 %.

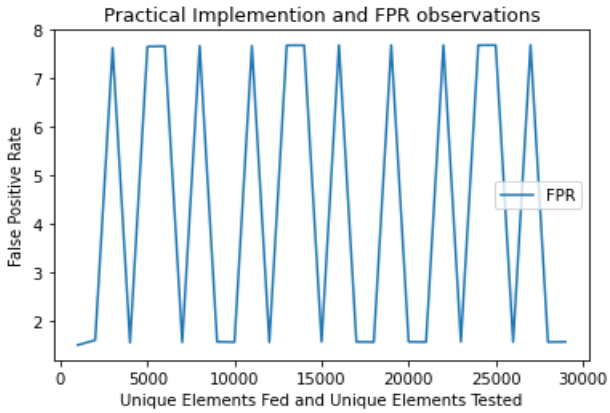

Fig. 5. FPR for 180823 bits size.

5.2. Using Memory Size of 97146 Bits in Improvised Bloom Filter

Figure.6 shows the practical implementation and FPR observations when taking 180823 bits size. In the graph, the X-axis indicates the unique elements fed and unique elements tested, and Y-axis indicates the false positive rates (FPR). The

FPR initialized as 20.0 %, and the Practical observation average: was 10.20874612312953 %.

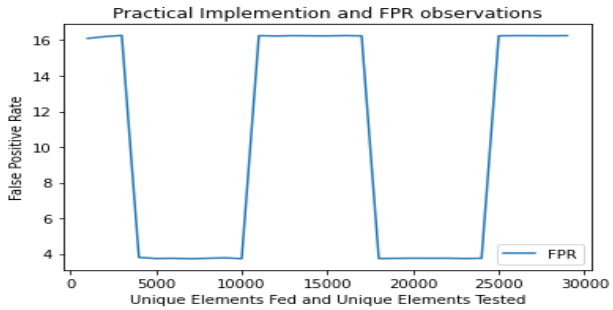

Fig. 6. FPR for 97146bits size.

5.3. Using Memory Size of 83677 Bits in Improvised Bloom Filter

Figure.7 shows the practical implementation and FPR observations when taking 180823 bits size. In the graph, the X-axis indicates the unique elements fed and unique elements tested, and Y-axis indicates the false positive rates (FPR). The FPR initialized as 25.0 %, and the Practical observation average: was 14.196855258603863 %.

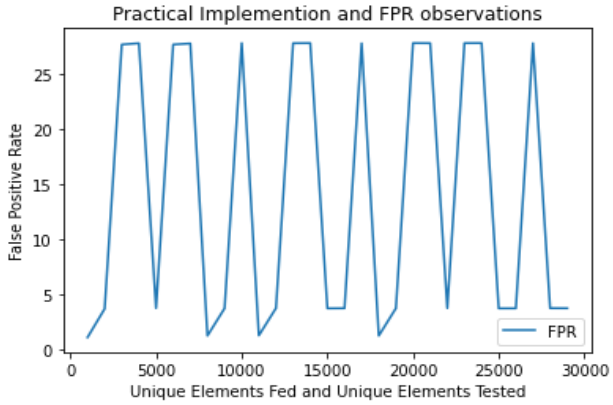

Fig.7. Practical observation average at 83677 bit size**.**

6. Conclusion

This paper created a the decentralized access and data storage system for cloud storage using the improvised bloom filter encryption and Distributed Hash Tables (DHT) framework for data stored in a decentralized manner. The suggested access control method aids in solving the multidimensional authorization problem in not just the data sharing system, but also in the data-sharing. The proposed framework eliminates the need for centralized storage since the upward directions of the monitoring mechanisms are returned via writing access data stored on the blockchain with permissions. The tests were conducted with Ethereum technology, which is a decentralized medium that can be used to define peer-to-peer networks, and the program is executed and written in Colab notebooks. The algorithm's false positive rate has been calculated using 3 variations of data sizes namely 180823 bits, 997146 bits as well as 83677 bits. The focus of our research is on improving the efficiency of access control decision-making by optimizing administration of algorithms.

References

1. K. Gai, Y. Wu, L. Zhu, M. Qiu, and M. Shen, “Privacy-preserving energy trading using consortium blockchain in smart grid,” *IEEE Transactions on Industrial Informatics*, 15(6), 2019, pp. 3548–3558.
2. S. Homayoun, A. Dehghantanha, R. M. Parizi, and K.-K.-R. Choo, “A blockchain-based framework for detecting malicious mobile applications in app stores,” in *Proceedings of the IEEE Canadian Conference of Electrical and Computer Engineering*, Edmonton, AB, Canada: IEEE, 2019, pp. 1–4.
3. X. Li, Y. Niu, L. Wei, C. Zhang, and N. Yu, “Overview on privacy protection in bitcoin,” *Journal of Cryptologic Research*, 6(2), 2019, pp. 133–149.
4. H.-N. Dai, Z. Zheng, and Y. Zhang, “Blockchain for internet of things: a survey*,” IEEE Internet of 6ings Journal*, 6(5), 2019, pp. 8076–8094.
5. X. Wang, X. Zha, W. Ni et al., “Survey on blockchain for internet of things,” *Computer Communications*, 136, 2019, pp. 10–29.
6. J. Bernal Bernabe, J. L. Canovas, J. L. Hernandez-Ramos, R. Torres Moreno, and A. Skarmeta, “Privacy-preserving solutions for blockchain: review and challenges,” *IEEE Access*, 7, 2019, pp. 164908–164940.
7. K. Gai, Y. Wu, L. Zhu, M. Qiu, and M. Shen, “Privacy-preserving energy trading using consortium blockchain in smart grid,” *IEEE Transactions on Industrial Informatics*, 15(6), 2019, pp. 3548–3558.
8. M. Conti, E. Sandeep Kumar, C. Lal, and S. Ruj, “A survey on security and privacy issues of bitcoin,” *IEEE Communications Surveys & Tutorials*, 20(4), 2018, pp. 3416–3452.
9. Y. Ren, Y. Liu, S. Ji, A. K. Sangaiah, and J. Wang, “Incentive mechanism of data storage based on blockchain for wirelesssensor networks,” *Mobile Information Systems*, 2018, 2018, pp. 6874158.
10. S. Han, Z. Xu, and L. Chen, “Jupiter: a blockchain platform for mobile devices,” *in Proceedings of the IEEE 34th International Conference on Data Engineering*, Paris, France, 2018, pp. 1649–1652.
11. Z. Guan, G. Si, X. Zhang et al., “Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities,” *IEEE Communications Magazine*, 56(7), 2018, pp. 82–88.
12. F. Gao, L. Zhu, M. Shen, K. Sharif, Z. Wan, and K. Ren, “A blockchain-based privacy-preserving payment mechanism for vehicle-to-grid networks,” *IEEE Network*, 32(6), 2018, pp. 184–192.
13. K. Panetta, “Top trends in the Gartner hype cycle for emerging technologies,” 2017, https://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-tec hnologies-2017.
14. Z. Zheng, S. Xie, H. Dai, and H. Wang, “An overview of blockchain technology: architecture, consensus, and future trends,” in *Proceedings of the IEEE International Congress on Big Data*, Boston, MA, USA, 2017, pp. 557–564.

A. M. Antonopoulos, *Mastering Bitcoin: Programming the Open Blockchain*, 2nd edition, O'Reilly, Media, CA, USA, 2017.
15. S. Singh, P. K. Sharma, S. Y. Moon, and J. H. Park, "Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions," *Journal of Ambient Intelligence and Humanized Computing*, 2017, pp. 1–18.
16. P. Treleaven, R. Gendal Brown, and D. Yang, "Blockchain technology in finance," *Computer*, 50(9), 2017, pp. 14–17.
17. L. Zhu, F. Gao, M. Shen et al., "Survey on privacy preserving techniques for blockchain technology," *Journal of Computer Research and Development*, 54(10), 2017, pp. 2170–2186.
18. M. Crosby, P. Pattanayak, S. Verma, and V. Kalyanaraman, "Blockchain technology: beyond bitcoin," *Applied Innovation*, 2, 2016, pp. 6–10.
19. 20.R. Dennis and G. Owen, "Rep on the block: a next generation reputation system based on the blockchain," in *Proceedings of the 10th International Conference for Internet Technology and Secured Transactions*, London, UK, 2015, pp. 131–138.