International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2023/218509 Volume 10, Issue 1, January – February (2023)

RESEARCH ARTICLE

Secure Storage and Data Sharing Scheme Using Private Blockchain-Based HDFS Data Storage for Cloud Computing

Gaurav Shrivastava
Department of Computer Science and Engineering, SAGE University, Indore, Madhya Pradesh, India.
gaurav2086@gmail.com

Sachin Patel
Department of Computer Science and Engineering, SAGE University, Indore, Madhya Pradesh, India.
drsachinpatel.sage@gmail.com

Received: 19 November 2022 / Revised: 30 December 2022 / Accepted: 07 January 2023 / Published: 26 February 2023

Abstract – The storage of a vast quantity of data in the cloud, which is then delivered via the internet, enables Cloud Computing to make doing business easier by providing smooth access to the data and eliminating device compatibility limits. Data that is in transit, on the other hand, may be intercepted by a man-in-the-middle attack, a known plain text assault, a selected cypher text attack, a related key attack, or a pollution attack. Uploading data to a single cloud might, as a result, increase the likelihood that the secret data would be damaged. A distributed file system extensively used in huge data analysis for frameworks such as Hadoop is known as the Hadoop Distributed File System, more commonly referred to as HDFS. Because with HDFS, it is possible to manage enormous volumes of data while using standard hardware that is not very costly. On the other hand, HDFS has several security flaws that might be used for malicious purposes. This highlights how critical it is to implement stringent security measures to make it easier for users to share files inside Hadoop and to have a reliable system in place to validate the shared files' validity claims. The major focus of this article is to discuss our efforts to improve the security of HDFS by using an approach made possible by blockchain technology (hereafter referred to as BlockHDFS). To be more precise, the proposed BlockHDFS uses the Hyperledger Fabric platform, which was developed for business applications, to extract the most value possible from the data inside files to provide reliable data protection and traceability in HDFS. In the results section, the performance of AES is superior to that of other encryption algorithms because it ranges from 1.2 milliseconds to 1.9 milliseconds. In contrast, DES ranges from 1.3 milliseconds to 3.1 milliseconds, three milliseconds to 3.6 millimetres, RC2 milliseconds to 3.9 milliseconds, and RSA milliseconds to 1.4 milliseconds, with data sizes ranging from 910 kilos.

Index Terms – Cloud Computing, Hadoop Distributed File System, Blockchain, Authenticity, Data Security, DES, AES.

1. INTRODUCTION

Over the last decade, research consortiums have achieved significant technical developments to adopt Data-sharing methodologies. Collaboration with others and making informed choices are two ways research-based activities might become more effective. The exchange of data is the first necessary step toward deriving the possible advantage from advances in research [1]. However, it is also essential to be aware of the "three W's" for sharing, which are "what," "when," and "where." Before beginning the data-sharing process, these questions must be resolved to everyone's satisfaction.

Cloud servers, a kind of centralised storage, are responsible for storing excessive data. One of the many hazards connected with centralised authorities is the possibility of a single point of failure. Third parties are brought in to offer data backups so that incidents like this may be avoided [2]. A blockchain gives trust and transparency, which helps remove the need for a third party to build a trust-based paradigm. Decentralised storage is a system that allows data to be stored independently on separate network nodes in the form of a distributed ledger. This may be accomplished via the use of a distributed ledger. Decentralised storage is a solution that was developed by blockchain technology. The issue is that network nodes have a limited capacity for both storage and computation. Interplanetary File System (IPFS), an architecture based on peer-to-peer communication, has been adopted for this purpose [3]. There is no possibility of failure at a single site [4]. It is very much like web3, but with some significant differences. It operates like a bit torrent and can perform content addressing [5]. Storing data on IPFS, a decentralised network, and guarantees that it will always be available when required.

RESEARCH ARTICLE

Hadoop is an open-source distributed file system, often known as HDFS [6,7], and is one of the business world's most well-liked choices for the batch processing of enormous volumes of data. [6,7] It has a stellar reputation thanks to its low latency and fast throughput for data access. There are various native file systems besides HDFS, such as Ceph [8], GPFS [9], and Hydra [10]. HDFS comes preconfigured with the MapReduce programming framework. Users may use one or more files while working with MapReduce to map and reduce (sort) the data that is included inside those files to produce the required output. However, to finish MapReduce [11] tasks, tiny amounts of Java code need to be written.

After MapReduce has finished reading a file from an input directory, it may be told to generate the required output in a separate output directory. HDFS may have been successful because of the way it was designed. While HDFS takes its cues from the UNIX file system, it differs because the information is spread over several drives rather than a single one. HDFS also offers the extra advantage of operating on affordable, commonly accessible hardware. Hadoop Distributed File System is the foundation upon which the powerful free and open-source framework known as Apache Hadoop is constructed (HDFS).

Hadoop was developed to create a system with high throughput and resistance to failure. Data may be loaded into HDFS in either the command line or application programming interfaces (APIs). Both the command line and the API contain a variety of commands that may be used to carry out a range of file activities, and if a remote connection is required, SSH may be used to establish it. To mention just a few of HDFS's features, users may create, read, delete, rename, list, and check the status of files. After HDFS is used to store data, several ecosystem frameworks, like Spark [12], Hive (https://hive.apache.org/) [13], and HBase [14], may be used to analyse the data. Remember that ecosystem apps often use the Hadoop Main API to access and manipulate data stored in HDFS.

Integrating an encryption technique into IPFS's hashes of uploaded data is how the distributed file system (IPFS) ensures that data is secure. In a later step, the owner encrypts these hashes by using the Shamir secret sharing (SSS) [10] technique. This scheme breaks the hash up into an n-number of encrypted shares. The smart contract is where the encrypted shares are kept safe.

The consumer initiates the process by submitting a request to view the data, which is subsequently validated using digital signatures. After the consumer has successfully downloaded the data using the decrypted shares, they must register reviews regarding the data using the review system. An incentive, which is the reimbursement of 10% of the real money placed by the client, has been proclaimed to encourage consumers to leave reviews and register for the contest.

The remaining portions of the paper are structured as follows: The current state of the art is discussed in Section 2, and a proposal based on the Ethereum blockchain is presented in Sections 3 and 4. Section 4 offers implementation details and simulation results. In the latter part of the work, the conclusion may be found in Section 5.

2. LITERATURE WORK

Data security issues from storing information over several clouds are the topic of this article, which proposes a method called Proficient Security over Distributed Storage to address these issues (PSDS). PSDS classifies the information as either normal or sensitive. The private information is also partitioned into two parts. Data is typically encrypted and uploaded to a single cloud; however, with this setup, individual components and the cloud are encrypted and distributed among several clouds. Sensitive information from many clouds is pooled when encryption is complete. Evidence has been shown that the PSDS is secure against a wide variety of attacks, such as related key attacks, pollution attacks, chosen ciphertext attacks, and attacks based on previously known plain texts. These results came from testing the PSDS against a variety of threats. Compared to STTN and RFD, PSDS encrypts data significantly faster due to its shorter computation time. [15]

A secure slicing-based resource orchestration (SS-RO) method has been developed to reduce the impact of slice-initiated attacks by optimising for low latency and low resource consumption. In this procedure, the interslice orchestration result is obtained by the Benders decomposition, while the interslice orchestration solution is obtained via the quadratic transformation technique. Both of these approaches will be discussed in detail below. The findings of the experiments show that the suggested SS-RO algorithm beats the baseline approaches in terms of the percentage of attacking jobs it accepts, the amount of energy it consumes, and the quantity of work it gets via the system. [16]

Electronic medical records, often known as EMRs, are kept at several different institutions and managed by a cloud service provider. Patients, who are the genuine proprietors of their private and sensitive EMRs, might lose track of them. To guarantee that all components of smart health-care systems can share electronic medical records (EMRs), this article aims to construct an access control framework that uses blockchain technology and smart contracts based on distributed ledger technology. For the objectives of user authentication, access authorisation, the detection of improper conduct, and the termination of access, we recommend using four smart contracts. After being encrypted using ECC and EdDSA, electronic medical records (EMRs) are kept in the cloud, and their hashes are published to a blockchain. A private Ethereum network is used to analyse real-time smart health care access control architecture. [17]

RESEARCH ARTICLE

The Internet of Things affects everything. IoT enables remote health monitoring. Medical data is plentiful. IoT has boosted volume. To have useful health data, we must preserve them effectively. We propose a cloudlet-enabled IoT e-health framework. This e-Health platform makes real-time cloudlet data retrieval easy. We design a healthcare data management system to store and handle end-user requests. NoSQL stores patient data. The proposed model analyses data transmission time, energy utilisation, query response time, and packet loss. We proved our model's superiority by comparing its findings to those competing for cloud-based e-Health systems. [18]

IoT technologies like smart meters, appliances, and grids may improve energy efficiency and customer service. Under standard AIoT paradigms, users' IoT energy data must be transported to a central repository (such as the cloud or an edge device) for knowledge extraction. Data exploitation and privacy issues may result. They provide an AIoT solution that works with edge clouds to share energy usage data in smart grids safely. Simulations show that the suggested strategy may increase communication and motivate EDOs to upgrade their local models. [19]

Significant changes have been made to load balancing outsourced in the cloud. The deploying blocks use a design for a distribution record that is shared across all servers to send data in a time-consent fashion without interfering with the usual functioning background modules of the record base a state machine. External data may be integrated and analysed. With cloud load balancing, several servers, networks, or individual PCs may share the processing of a single task. DopCloud optimises the packet flow routing in the cloud by analysing it and traversing several stack layers at a microsecond scale. The server and client codes must be updated with each deployment to maintain reliable load balancing and cloud backups. Due to an abruption in data storage, electronic health record (EHR) systems in the medical profession were forced to cease operations, resulting in data loss. Adjustments are piled to control the cloud network, making those controls available for the diagnostic data load. Here, RIFT and VNF cloud provide a more trustworthy protocol that promises to protect massive data loads. It is the packet-flowing route optimisation (DOPER) system that does the bulk of the planning for the containerisation of the state machine, and no communication is allowed between DopCloud peer nodes until all contracts are made accessible to all nodes. [20]

Cloud networks safeguard the processing of data and its transmission to users with appropriate permissions. "cloud computing" refers to a data storage and management model that uses several computers in different parts of the world. There is a need for load balancing. Within the scope of this research project, a hybrid heuristic mathematic algorithm (HHMA) is proposed for IaaS computing networks to resolve issues with resource allocation. To speed up the design process and download of files, improved K-means clustering was used to divide the cluster into many little parts (heuristic). MCSO uses real-time constraints to determine a determined node's load ratio. Following the execution of the MCSO algorithm, the optimal value for the complete evaluation is determined and used to allocate controller nodes to storage nodes. Two strategies distribute request data to relevant nodes for efficient processing. In comparison to earlier approaches, this one uses far fewer resources. The simulation findings imply that the suggested technique would lessen the burdens on memory, reaction time, and network overhead. Investors in cloud computing may benefit from the use of the recommended heuristic method. [21]

The dawn of the information age in China resulted in a rise in the amount of documentation produced by end users. This came with it the difficulty of finding out how to store enormous volumes of data in a secure and straightforward manner. The technology of the cloud could solve this issue. People have been able to reduce their demand for hardware by adopting cloud storage, yet there is still a worry over hardware security. This work discusses dividing and merging source documents, encrypting the division table, recovery, and backup. [22]

WukaStore can provide a huge data storage solution that is scalable, adaptable, and reliable by combining non-volatile storage, such as that provided by the cloud, with volatile storage, such as that which is harvested from unused storage space on desktop computers via the internet. By customising a number of different storage methods, WukaStore offers storage that is efficient and affordable. In order to investigate how to guarantee the high availability and durability of WukaStore, we use trace-driven simulations. The open-source Big Data middleware BitDew serves as the foundation for WukaStore. On the experimental platform that France has developed, Grid'5000, we evaluate how well WukaStore performs. [23]

Customers preserve several copies of cloud-hosted data to increase its availability and durability. Multi-copy data's integrity is ensured via PDP methods. All PDP copies are stored on a single cloud server. This shouldn't be duplicated. Many PDP protocols need expensive and insecure public key infrastructure (PKI). We present an identity-based PDP system that stores data on various cloud servers to increase safety and efficiency. [24]

The cloud is being used by many corporations, academic institutions, government agencies, and other organisations because of its cheap initial cost, scalability, and other advantages. The cloud offers a great many advantages, but it also has a great many disadvantages. Data protection is given top priority by both information security and cloud computing. This problem can be solved in a few different

RESEARCH ARTICLE

ways. Because the already available solutions have not been subjected to a comprehensive analysis, it is required to conduct research, classify, and evaluate the work that has already been done to determine whether or not it meets the requirements. This article compares and analyses the primary cloud data sharing and protection methods. The discussion of each specific method contains the following topics: data security functions, prospective and novel solutions in the field, workflow, accomplishments, scope, gaps, future directions, etc. In addition, a comparative analysis of the approaches has been presented. After that, a discussion of the applicability of the techniques follows, after which research gaps and potential future projects are outlined. The writers of this paper believe that it will inspire the next generations of researchers to investigate the subject. [25]

Users may easily back data to the cloud and save archives with MCS. Here, we focus on how frequently data is accessed to create a private, secure, and effective cloud storage solution for mobile devices. As the central mechanism of the mobile cloud storage system, we propose an OSU protocol. The client's computation and communication overheads are reduced since they only have to construct a tiny encrypted vector to get encrypted data from the cloud and update it. In contrast to prior research, our approach provides a fine-grained data structure with a tiny item size, requiring just a few additively homomorphic operations on the client side and a constant communication cost. Because of the usage of verification chunks, our technique is immune to attacks from malicious clouds. Our method is more efficient when evaluating client and cloud workloads than the current storage solutions. [26]

Access control methods in cloud storage are becoming important as cloud computing evolves. In business, the Chinese Wall is a tried-and-true solution for dealing with the CoI issue. The capacity to store conflict-proof data in the cloud might be useful. Users' interests, investing choices, and the access control mechanism may reveal other personal information since it does not provide anonymity. Problems arise while building the Chinese Wall without compromising users' privacy. Cloud storage is implemented utilising the Chinese Wall technique to protect user access patterns.

The C-Wall protocol will be discussed next once the tree-based Chinese Wall access control has been introduced. Protecting the user's anonymity, our C-Wall is impenetrable to hostile actors and may be built anywhere in the world. We further extend C-Wall to privacy-protecting cloud storage with C2-Wall, which keeps all the advantages of C-Wall and stops "honest but inquisitive" cloud servers from accessing private data. These two additions represent significant advances in the discipline. When it comes to our C2-Wall, we test and check it thoroughly. According to experiments, it has real-world applications. [27]

Computing on the cloud presents an opportunity for storage solutions. Concerns about the safety of cloud storage might limit its expansion. Cloud-stored data is vulnerable to malicious changes and data loss. Recent research outlines a three-layer fog server architecture for cloud storage. Hash algorithm and Hash-Solomon code are modified. It didn't improve the ability to identify changes or recover lost data, but it lost less to cloud servers. This article describes fog-centric secure cloud storage to prevent unauthorised data access, modification, and deletion. The recommended system hides data using XorCombination and XorCombination. Block Management outsources Xor-Combination results to prevent malicious retrieval and aid data recovery. A hash-based technique improves change detection. The system's security is analysed. Experiments show the proposed approach is faster than existing options. [28]

There is a growing need for object storage in the cloud to manage massive amounts of large binary objects (BLOBs), which include movies, images, and documents. The vast majority of the data encoding strategies being used are not ideal for the architecture of cloud object storage, despite the way that several companies and organisations want to utilise public cloud object storage services like Amazon Simple Storage Service (S3).

In this investigation, we provide a technique called dynamic extreme erasure encoding, often called DexEncoding. Its purpose is to enhance client utility by dynamically optimising encoding sites between gateway and storage servers in a cloud environment that varies over time. The utility gauges the degree to which customers are happy with the speed and fairness of data storage. To efficiently relieve resource limits, DexEncoding can adapt to the availability of the network, processing, and storage resources, as well as storage requests. Simulations driven by real-world measurements demonstrate that DexEncoding provides higher levels of customer satisfaction than other cutting-edge object storage systems. [29]

IoT and smart diagnostic implants improve medical systems by accessing remote patient data and screening for health risks. IoT medical gadgets monitor patients' health and transfer data to the cloud for doctors. When it comes to the exchange of data and outsourcing, the privacy and secrecy of patients' electronic medical records are very important considerations that must be considered. In this study, a solution for securing data storage and regulating access is proposed for use in intelligent healthcare systems. This article includes a password, a psychometric slot enabled by deep learning, and an evaluation based on deep learning. The proposed solution has undergone testing and demonstrated that it is feasible in comparison to the existing access control systems. [30]

RESEARCH ARTICLE

Together with the surge in the popularity of cloud computing in general, the demand for cloud-based block storage services has increased. Performance optimisation becomes more difficult when resource demand is not constant throughout a cloud block storage system. [31]

Understanding how to verify the integrity of data stored in the cloud is becoming an increasingly vital skill. ID-based proved data possession (PDP) is an auditing method for cloud storage that does not need a certificate and does not require user data. Cloud users must offshore data blocks, authenticators, and a small file tag to use an existing ID-PDP system. Additionally, bilinear pairing and elliptic curve cryptography are required. These drawbacks would increase storage, connectivity, and computing expenses, which was not anticipated for users of cloud services with limited resources.

This is made possible by the absence of substantial cryptographic processes, contributing to the protocol's overall effectiveness. The recommended protocol may give additional helpful functions thanks to its usage of primitive replacement. The suggested protocol cannot be falsified, can be found, is accurate, and is detectable. In conclusion, we provide both theoretical and experimental data to validate the recommended technique. [32]

CDS is a popular cloud-based solution because of its cheap cost and excellent efficiency in disseminating and sharing content with end-users, partners, and insiders. By duplicating commonly used resources, this service increases both accessibility and I/O speed. This adds more work for the network and storage facilities. This study proposes a three-pronged strategy for enhancing the effectiveness of I/O operations in the cloud and increasing CDS use. [33]

As cloud storage services grew more common, there was an increase in worry over the integrity of data that was outsourced and stored on servers that could not be trusted. Provable data possession (PDP) protects the integrity of cloud-stored data by demanding the cloud server to prove the data hasn't been updated or deleted without giving users access to the actual data. ID-P33DP is an identity-based, privacy-preserving data possession mechanism for secure cloud storage. A cloud user must utilise an outsourced file and a global parameter to create ID-P33DP authenticators. Any third-party auditor (TPA) may ascertain whether an outsourced file was retained in its original state by validating homomorphic authenticators. ID-P33DP supports the aggregation of user-generated identity-based homomorphic authenticators using RSA. The cloud may aggregate identity-based homomorphic authenticators to create a data possession proof that verifies data integrity. A zero-knowledge proof may prevent TPA data leaks. RSA proves ID-soundness, P33DPs' privacy, and TPA's privacy. Cross-user aggregate verification reduces TPA's computational and communication cost [34].

2.1. Problem Statement

The sharing of sensitive data in the cloud still confronts significant problems, including achieving data privacy and maintaining lightweight operations at mobile terminals with limited resource availability.

3. CONTRIBUTION

In the foundational work, the Ethereum blockchain is used, outlining transaction blockchain and having overheads of every transaction, both of which may be elicited by using a private blockchain. Base work im Ipfs is a file system that is utilised. However, it is sluggish and difficult to make storage process changes. HDFS is superior to ipsf because it provides data distribution over several clusters, making it more reliable and secure. Proof of authority (POA) is used in the foundational work. In contrast, proof of identity (POI) and proof of capacity (POC) are employed in the present one since the data manipulator plays a crucial part in data handling throughout the transfer of information from the uploader to the blockchain.

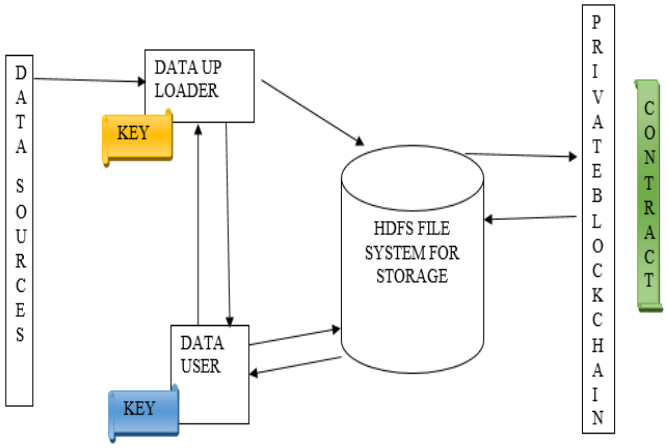

Figure 1 Architecture

Figure 1 shows how data is moved from its sources into the HDFS system and encrypted using the data's key during the upload process. From there, the data is moved into a private blockchain network, which is authenticated using the network's contract of consensus mechanism. When a data user requests data from the blockchain, the request is first routed through the HDFS storage system using the user's key. The file system then verifies the user's identity before, in response to the user's request, it pushes the data to the data user. It updates the blockchain with transaction information. This procedure follows the same pattern as the uploader.

3.1. Proposed Algorithm

1. Input: z
2. { Step 1: Upload IoT data to HDFS}
3. Bytes64[DataList]

RESEARCH ARTICLE

4. Maps (Bytes64 H⇒ Data) DataMap
5. Maps (Bytes64 H⇒ String) hdfs.io

6. Event confirm Hierarchy() // call event confirmation

7. Event confirmFileIndexID() // file index updated
8. Function SendDataToFilesystem(z): / transferring files

9. $Set_{DO} \leftarrow msg.sender_;$

10. Require ← DU_meta_Mask_Add
11. PrivateAccAdd = await
12. web.Pt.get_accounts()
13. SaveToHDFS awaithdfs.add ← buffer
14. GenerateHash this.setstate ←hdfshash

15. StoreHash AgainstId ← Store_hash._methods.send_hash(_this.state.hdfs)
16. Transcation ← this. set_State({transcation_hash}_)
17. }
18. Step 2:{

19. Rewardget ← DOAssetsAvailable

20. if DO_Assets_Available < tx. amount _then
returns_Error_message
_Else

21. _for i ← 0 _to_ tx.amount -1 _do
22. Rewardget[i]. DO ← asset. DO
23. Update _Asset_Registery}
24 Step 3: {
_Emit _event_ of_ sharing
25. _Update _asset_ status // status updating

26. Create _logs // logs creation

27. Return _Sharing _successful

Algorithm 1: Data Storage and Sharing Through an Incentive-Based HDFS

Algorithm 1 shows that after data is uploaded to HDFS, the algorithm generates a hash, stores it in HDFS, and updates the hash in the blockchain. Finally, the sender sends a message, the data user creates a mask, and both accounts are confirmed before the data sender saves the data to HDFS. Once data transactions are recorded, the system is ready to exchange data. Step 3 generates logs anytime data is used or uploaded by any data uploaded or data user, and if the transaction fails, the data sender gives an error message.

Input: user management information

Output: authorised user information

1. User handlers

2. AddNewUser(){raise an exception if the msg.sender is not DO;
3. if msg.sender is NewUserAddr then
4. RegisteredUserList [NewUserAddr]
5. return true;
6. else
7. [NewUserAddr] exist;
8. 1return false}

9. UpdateDetails(){if_msg. sender_ is_ NOT then_ 4throw;

10. If _map _authorized__Users_list[i] ← false_ then return_ false;

11. else

12. Map _authorized_Users_list[Update_User] ← true; return true}

Algorithm 2: Features Like Add New User, Update Details, Delete File and User for Operation

In algorithm 2 shows the details in below: Add a new user: this algorithm 2 produces a user profile to add data uploaded or data users over the system. It gives every data manipulator a unique username and password. This algorithm 2 updates the user's information and can also update the data details across the system. Additionally, it generates logs of every modification to maintain track of earlier versions. Delete files: these algorithmic 2 procedures are used to delete files from the file system. They also delete user accounts from the system's access information, enabling the system to remove users successfully. File from side systems; the most important point to remember is that if any file from a user is sent over the network, then the user information for that specific user is not removed from the network.

4. HARDWARE AND SOFTWARE REQUIREMENTS

4.1. Hardware and Software

A full-screen, touch-enabled IPS panel measuring 14 inches in size can be folded into the form factor of a tablet (the x360 Touchscreen 2-in-1). Python and Java were used for this endeavour, and a laptop included a 10th-generation Core i7-10510U processor and a 512GB solid-state drive. The operating system of option was Windows 10 Home 64 Bit. Different kinds of processors include: The L3 cache is 8 MB, and the clock speed is 1.6 GHz. There are four processing cores, and the base clock may vary anywhere from 1.6 GHz to 4.9 GHz, owing to the Turbo Boost Technology developed by Intel. This system also has HD Audio and Intel Iris Plus Graphics as standard equipment. HP makes the HD True Vision camera accessible to its customers. NumPy, Pandas, SciPy, PyTorch and Plotly, Keras, and OpenCV-python are only a few Python libraries used in the process. Cloud Analyst, Big Data frameworks.

5. RESULT

In this section, we have to explain the result in different environments in tables 1 to 6 and figures 2 to 3.

RESEARCH ARTICLE

Programmes known as distributed apps (dApps) are created in languages such as Go, Java, or Node.js [36]. To be more explicit, the nodes may be clients that propose transactions and broadcast them to peers for ordering, peers in charge of maintaining the ledger and its state or ordering service nodes that determine the order in which transactions occur. The latter party is not responsible for carrying out either the execution or the validation procedures. The fabric uses smart contracts, sometimes called chain codes, to construct the application logic. Using this method comes with several potential problems, one of which is that it has the potential to render the system inoperable if one-third or more than one-third of the validators are not currently online. The performance of the Ethereum blockchain is much higher than that of other blockchain networks, as seen in Table 1.

Table 1. To Test the Costs Associated with IPFS Smart Contracts, 1 Gas Unit is Equivalent to 2 Gwei, and 1 ETH is Equivalent to $3,590.

IPFS Functions	Transaction Cost (Gwei)	Execution Cost (Gwei)	Actual Cost (ETH)
IPFS data storage contract create	302145	278965	0.000737
IPFS data sharing contract create	285634	198563	0.000421
DeleteFile	37245	23451	0.000063
AddUser	58632	39562	0.000063
Update User	37456	22457	0.000022
Remove User	24516	11547	0.000019

Table 1 presents the gas usage and associated costs calculated for smart contracts. The Ethereum platform hosts a smart contract that serves as the foundation of the review system. For the deployment of any contract, gas is necessary. The designer has set the maximum amount of gas at 2,000,000, and this restriction must be adhered to at all times. When a contract is being carried out for the very first time, a higher level of intensity is expected from all parties involved. The final result is that the limit is raised to a higher number, the specific nature of which is determined by the size of the blocks and the charge that the miner is required to pay.

The amount of gas used is considered as a cost component throughout the execution and the transaction phases of the process. The terms "transaction gas" and "execution gas" refer to the two distinct kinds of gas used. The amount of gas required to carry out any activity on the blockchain network is referred to as the transaction cost.

In contrast, the computational price required to carry out the smart contract is the execution cost. The transaction cost and the execution cost are both referred to as the cost of carrying out the smart contract. We first needed to establish the computational cost limit that would apply to the beginning and end of transactions on the blockchain before we could successfully execute smart contracts.

Table 2. Test of the System's Security Costs (GasPrice Equals 3 Gwei, 2 ETH Equals 3,590 USD).

Security Parameters	Size (bytes)	Gas Used	Actual Cost (ETH)
Master Key (MK)	124	18523	0.000031837
Secret key (SK_{DU})	842	65237	0.000086523
Cipher Text (CT_F)	463	36985	0.000071423
Proof of Work (PoW)	--	853624	0.001235
Proof –of-Authority (PoA)	--	485236	0.001236

Following a BASE 64 encoding approach, we converted each data item to the JSON format to create results that were simpler to understand. The experiment results are summarised in Table 2, which provides an overview of some of the costs associated with the smart contract.

Table 3 Uploading Data to IPFS and Uploading of Ciphertext to the Ethereum Blockchain Network

Key	Value
Status	0x1 Successfully mined and executed transaction
Block	8965475 (49851)Block Confirmations
Timestamp	9 days 32 hrs. ago (Aug − 08 − 2022 08: 24: 02 AM + UTC)
From	0us84jkd12b3dak68s32k14a0e9b687 1acf6b1720
To	[Contract 0fs84586b9s98f2e07 1f27c6578e5d6f9568ff48e9 Created]
Transaction Hash	0x24d8vd21456f5s3fffgh541236jh87 4rty6hgderf3214hjkr98hufrt3214hgf 68745ygbn945217lkas964gytr
Ethereum Contract address	0x54asdfg7854rtghf6321bnmvc541v gy12332145gfbv9874
Data Hash in IPFS	Data)Storage. Send_Data_TOIPFS(by 0xdfgg32145asdf632145hytre69874 5oplkf32147gfvb5641

RESEARCH ARTICLE

Input	0xdnf698fssf56rth236541nmkjjg9874586dfrtyuh321459mkloiurd178569ddfrthhf22365
Transaction_Fee	0.0005236545984532148 – Ether
Gas_Price	0.000000000123654789 – Ether – (1.56321478 Gwei)
Gas_{Limit} & Usage_by	865, 546 984 , 854 (100%)

The hash value was encrypted using encryption compliant with the Advanced Encryption Standard (AES-128), and the resultant encrypted value can be found in Table 3. Smart contracts were put through their paces for performance and functionality testing on the Rinkeby Testnet test network. The AES-encrypted ciphertext is then saved in a blockchain storage area that cannot be altered by anybody other than the smart contract's creator. Table 3 provides a thorough breakdown of the contract's actual execution, which is also available for your perusal.

Table 4 Comparative Analysis with Many Previous Studies

Reference	Blocchain	IPFS	IOT Data	Data Encryption	Data Sharing	Fine Grained Permission	Data Control	Key Word Search
[28]	√	√	×	√	√	√	×	×
[30]	√	×	√	×	√	×	×	×
[31]	√	√	×	×	√	×	√	√
[32]	√	×	√	√		×	×	√
[33]	√	√	√	×	√	×	√	×
[34]	√	×	√	×		√	√	×
[35]	√	√	√	√	√	√	√	√
Proposed	√	√	√	√	√	√	√	√

In addition, we spoke about some of the parameters that are a part of our framework and evaluated relevant work using those parameters. In Table 4, we compare our IoTChain model and many other models currently used for decentralisation, distributed data storage and sharing, data encryption, keyword search, verified results, and access control.

These comparisons are based on what we accomplished with our model of the IoTChain. In addition, it is of the utmost importance that these factors be included in the framework while preserving security and privacy. Our notation of "" for the tactic including this feature and "for the converse condition both clarify the meaning of the symbols' respective contexts. Table 5 shows that neither the two proposed methods, [28] nor [30], meet access controls, encryption, and searchable data requirements. Unlike the techniques [32], the systems [31] and [33] provide for off-chain data storage, interchange, and access control [34], and none of them supports keyword search. In addition, our system saves the data in IPFS, eliminating the possibility of the data being altered or lost within a cloud environment. In addition, we are the only ones with a plan that satisfies all requirements and is more appropriate for the present. In figure 2, the proposed work hdfs is used to store data as a file system which improves the data transfer rate and reduces the cost in data storage, sc cost takes 21036 and execution.

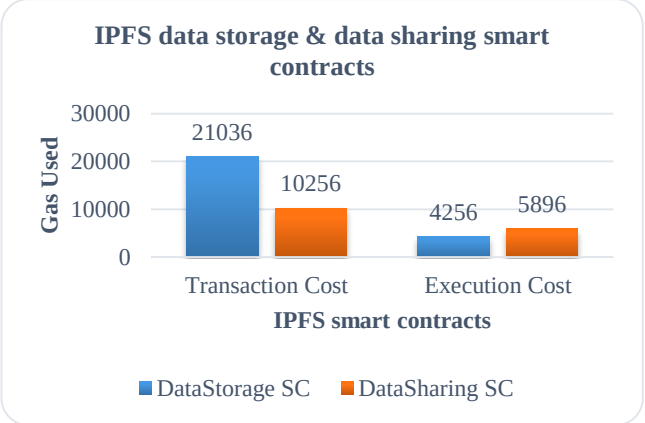

Figure 2 IPFS is a Decentralised Data Storage and Sharing Platform that also has Smart Contracts

In figure 3, all types of manipulation have some cost in the file system. This table represents the deleted file, the addition of the file, updating of the file and the removing file operation transaction cost and execution cost.

RESEARCH ARTICLE

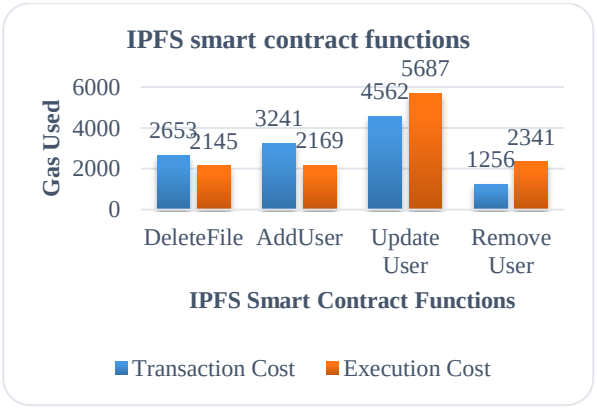

Figure 3 IPFS Smart Contract Functions

Table 5. A comparison of the Proof-of-Work, Proof-of-Assembly, Proof-of-Instruction, and Proof-of-Consensus Consensus Mechanisms in Terms of Gas Consumption and Time (ms) Taken

	PoW	PoA	PoI	PoC
DataStorage	924893	589483	1354	2317
DataSharing	513245	364324	896	2986

This table 5 represents how much gas is consumed in pow,poa,poi and poc ppi and poc both are significantly less gas consumption required as their data storage gas consumption is 1354,2317 respectively for both. Similarly, they consume less gas in data sharing as 896 and 2986 respectively.

Table 6 Typical TX Commit Period Using a Hybrid of Proof-of-Work, Proof-of-Authority, Proof-of-Importance, and Proof-of

	PoW	PoA	PoI	PoC
DataStorage SC	678.4	487.6	105	81
DataSharing SC	564.7	366.5	80	98

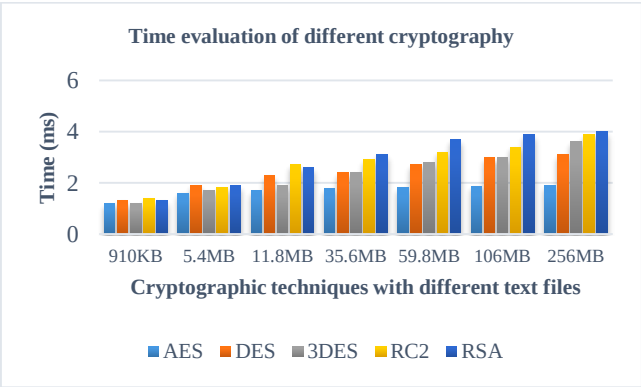

Figure 4 Time Evaluation of Different Cryptography

Table 6 explains about commit times of poa,pow,poi and poc so poi and poc in data storage having 105 ,81 ms respectively and in data sharing they have 80 ,98 ms respectively as commit time.

Figure 4 explains how various encryption algorithms take time while the size of the data increases; AES performs better than other encryption algorithms, ranging from 1.2ms to 1.9. In contrast, DES ranges from 1.3 to 3.1, 3DES ranges from 1.2 to 3.6, RC2 ranges from 1.4 to 3.9, and RSA ranges from 1.4 to 4 ms, as data sizes range from 910KB to 256 MB.

6. CONCLUSION

The storage of a vast quantity of data in the cloud, which is then delivered via the internet, enables Cloud Computing to make doing business easier by providing smooth access to the data and eliminating device compatibility limits. Data that is in transit, on the other hand, may be intercepted by a man-in-the-middle attack, a known plain text assault, a selected cypher text attack, a related key attack, or a pollution attack. Uploading data to a single cloud might, as a result, increase the likelihood that the secret data would be damaged. The Hadoop Distributed File System (HDFS) is a popular choice among distributed file systems for large-scale data processing on Hadoop and similar frameworks. We plan to enhance the safety of HDFS by using a method enabled by blockchain technology (hereafter referred to as BlockHDFS). More specifically, the proposed BlockHDFS uses the Hyperledger Fabric platform, which is designed for corporate usage, to make the most of the information inside files to provide trustworthy data security and traceability in HDFS. In the results section, the performance of AES is superior to that of other encryption algorithms because it ranges from 1.2 milliseconds to 1.9 milliseconds. In contrast, DES ranges from 1.3 milliseconds to 3.1 milliseconds, 3 milliseconds to 3.6 millimeters, RC2 milliseconds to 3.9 milliseconds, and RSA milliseconds to 1.4 milliseconds, with data sizes ranging from 910 kilos.

REFERENCE

[1] G. Kumar et al., "A Novel Framework for Fog Computing: Lattice-Based Secured Framework for Cloud Interface," in IEEE Internet of Things Journal, vol. 7, no. 8, pp. 7783-7794, Aug. 2020, doi: 10.1109/JIOT.2020.2991105.

[2] X. Liu, G. Yang, Y. Mu and R. H. Deng, "Multi-User Verifiable Searchable Symmetric Encryption for Cloud Storage," in IEEE Transactions on Dependable and Secure Computing, vol. 17, no. 6, pp. 1322-1332, 1 Nov.-Dec. 2020, doi: 10.1109/TDSC.2018.2876831.

[3] Benet, J. Ipfs-content addressed, versioned, p2p file system. arXiv 2014, arXiv:1407.3561.

[4] J. Wei, X. Chen, X. Huang, X. Hu and W. Susilo, "RS-HABE: Revocable-Storage and Hierarchical Attribute-Based Access Scheme for Secure Sharing of e-Health Records in Public Cloud," in IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 5, pp. 2301-2315, 1 Sept.-Oct. 2021, doi: 10.1109/TDSC.2019.2947920.

[5] H. Wang, L. Feng, Y. Ji, B. Shao and R. Xue, "Toward Usable Cloud Storage Auditing, Revisited," in IEEE Systems Journal, vol. 16, no. 1, pp. 693-700, March 2022, doi: 10.1109/JSYST.2021.3055021.

RESEARCH ARTICLE

[6] Apache Hadoop, URL, http://hadoop.apache.org, 2006.
[7] K. Shvachko, H. Kuang, S. Radia, R. Chansler, The hadoop distributed file system, in: 2010 IEEE 26th Sym- Posium on Mass Storage Systems and Technologies (MSST); 3–7 May 2010; Incline Village, NV, USA, IEEE, Piscataway, NJ, USA, 2010, pp. 1–10.
[8] S.A. Weil, S.A. Brandt, E.L. Miller, D.D.E. Long, C. Maltzahn, Ceph: a scalable, high performance dis- tributed file system, in: Proceedings of the 7th Symposium on Operating Systems Design and Implementation, OSDI '06; 6–8 Nov 2006; Seattle, WA, USA, USENIX Association, Berkeley, CA, USA, 2006, pp. 307–320.
[9] F. Schmuck, R. Haskin, Gpfs: a shared-disk file system for large computing clusters, in: Proceedings of the 1st USENIX Conference on File and Storage Technologies, FAST '02; 28–30 Jan 2002; Monterey, CA, USA, USENIX Association, Berkeley, CA, USA, 2002.
[10] C. Ungureanu, B. Atkin, A. Aranya, et al., HydraFS: A high-throughput file system for the hydrastor content-addressable storage system, in: Proceedings of the 8th USENIX Conference on File and Storage Technologies, FAST'10; 23–26 Feb 2010; San Jose, CA, USA, USENIX Association, Berkeley, CA, USA, 2010, pp. 225–239.
[11] J. Dean, S. Ghemawat, Mapreduce: simplified data processing on large clusters, Commun. ACM 51 (1) (2008) 107–113.
[12] M. Zaharia, R.S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M.J. Franklin, et al., Apache spark: a unified engine for big data processing, Commun. ACM 59 (11) (2016) 56–65.
[13] X. Chen, L. Hu, L. Liu, J. Chang and D. L. Bone, "Breaking Down Hadoop Distributed File Systems Data Analytics Tools: Apache Hive vs. Apache Pig vs. Pivotal HWAQ," 2017 IEEE 10th International Conference on Cloud Computing (CLOUD), 2017, pp. 794-797, doi: 10.1109/CLOUD.2017.117.
[14] M.N. Vora, Hadoop-hbase for large-scale data, in: Proceedings of 2011 International Conference on Computer Science and Network Technology; 24–26 Dec 2011; Harbin, China, IEEE, Piscataway, NJ, USA, 2011, pp. 601–605.
[15] F. Shahid, H. Ashraf, A. Ghani, S. A. K. Ghayyur, S. Shamshirband and E. Salwana, "PSDS–Proficient Security Over Distributed Storage: A Method for Data Transmission in Cloud," in IEEE Access, vol. 8, pp. 118285-118298, 2020, doi: 10.1109/ACCESS.2020.3004433.
[16] J. Tang, J. Nie, Z. Xiong, J. Zhao, Y. Zhang and D. Niyato, "Slicing-Based Reliable Resource Orchestration for Secure Software-Defined Edge-Cloud Computing Systems," in IEEE Internet of Things Journal, vol. 9, no. 4, pp. 2637-2648, 15 Feb.15, 2022, doi: 10.1109/JIOT.2021.3107490.
[17] A. Saini, Q. Zhu, N. Singh, Y. Xiang, L. Gao and Y. Zhang, "A Smart-Contract-Based Access Control Framework for Cloud Smart Healthcare System," in IEEE Internet of Things Journal, vol. 8, no. 7, pp. 5914-5925, 1 April1, 2021, doi: 10.1109/JIOT.2020.3032997.
[18] S. Sengupta and S. S. Bhunia, "Secure Data Management in Cloudlet Assisted IoT Enabled e-Health Framework in Smart City," in IEEE Sensors Journal, vol. 20, no. 16, pp. 9581-9588, 15 Aug.15, 2020, doi: 10.1109/JSEN.2020.2988723.
[19] Z. Su et al., "Secure and Efficient Federated Learning for Smart Grid With Edge-Cloud Collaboration," in IEEE Transactions on Industrial Informatics, vol. 18, no. 2, pp. 1333-1344, Feb. 2022, doi: 10.1109/TII.2021.3095506.
[20] S. Srinivasan, "Cloud load balancing: Blockchain deployment at integrated DopCloud synthesis on Healthcare data," 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), 2020, pp. 364-369, doi: 10.1109/ICSSIT48917.2020.9214084.
[21] G. Senthilkumar and M. P. Chitra, "A Novel hybrid heuristic-metaheuristic Load balancing algorithm for Resource allocationin IaaS-cloud computing," 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), 2020, pp. 351-358, doi: 10.1109/ICSSIT48917.2020.9214280.
[22] J. Cai, Y. Hu and Y. Li, "Research on the Method of Building a Secure Cloud Storage Platform," 2022 3rd International Conference on Electronic Communication and Artificial Intelligence (IWECAI), 2022, pp. 17-20, doi: 10.1109/IWECAI55315.2022.00011.
[23] B. Tang and G. Fedak, "WukaStore: Scalable, Configurable and Reliable Data Storage on Hybrid Volunteered Cloud and Desktop Systems," in IEEE Transactions on Big Data, vol. 8, no. 1, pp. 85-98, 1 Feb. 2022, doi: 10.1109/TBDATA.2017.2758791.
[24] J. Li, H. Yan and Y. Zhang, "Efficient Identity-Based Provable Multi-Copy Data Possession in Multi-Cloud Storage," in IEEE Transactions on Cloud Computing, vol. 10, no. 1, pp. 356-365, 1 Jan.-March 2022, doi: 10.1109/TCC.2019.2929045.
[25] Gupta, A. K. Singh, C. -N. Lee and R. Buyya, "Secure Data Storage and Sharing Techniques for Data Protection in Cloud Environments: A Systematic Review, Analysis, and Future Directions," in IEEE Access, vol. 10, pp. 71247-71277, 2022, doi: 10.1109/ACCESS.2022.3188110.
[26] J. -N. Liu et al., "Enabling Efficient, Secure and Privacy-Preserving Mobile Cloud Storage," in IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 3, pp. 1518-1531, 1 May-June 2022, doi: 10.1109/TDSC.2020.3027579.
[27] X. Li, T. Xiang, Y. Mu, F. Guo and Z. Yao, "C-Wall: Conflict-Resistance in Privacy-Preserving Cloud Storage," in IEEE Transactions on Cloud Computing, doi: 10.1109/TCC.2022.3171772.
[28] M. A. M. Ahsan, I. Ali, M. Imran, M. Y. I. B. Idris, S. Khan and A. Khan, "A Fog-Centric Secure Cloud Storage Scheme," in IEEE Transactions on Sustainable Computing, vol. 7, no. 2, pp. 250-262, 1 April-June 2022, doi: 10.1109/TSUSC.2019.2914954.
[29] K. Lee, J. Kim, J. Kwak and Y. Kim, "Dynamic Multi-Resource Optimization for Storage Acceleration in Cloud Storage Systems," in IEEE Transactions on Services Computing, doi: 10.1109/TSC.2022.3173333.
[30] G. Revathy, P. Muruga Priya, R. Saranya and C. Ramchandran, "Cloud Storage and Authenticated Access For Intelligent Medical System," 2022 6th International Conference on Computing Methodologies and Communication (ICCMC), 2022, pp. 53-56, doi: 10.1109/ICCMC53470.2022.9753765.
[31] C. Liang, L. Deng, J. Zhu, Z. Cao and C. Li, "Cloud Storage I/O Load Prediction Based on XB-IOPS Feature Engineering," 2022 IEEE 8th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), 2022, pp. 54-60, doi: 10.1109/BigDataSecurityHPSCIDS54978.2022.00020.
[32] Y. Yang, Y. Chen, F. Chen and J. Chen, "An Efficient Identity-Based Provable Data Possession Protocol With Compressed Cloud Storage," in IEEE Transactions on Information Forensics and Security, vol. 17, pp. 1359-1371, 2022, doi: 10.1109/TIFS.2022.3159152.
[33] V. J. Sosa-Sosa, A. Barron, J. L. Gonzalez-Compean, J. Carretero and I. Lopez-Arevalo, "Improving Performance and Capacity Utilisation in Cloud Storage for Content Delivery and Sharing Services," in IEEE Transactions on Cloud Computing, vol. 10, no. 1, pp. 439-450, 1 Jan.-March 2022, doi: 10.1109/TCC.2020.2968444.
[34] J. Ni, K. Zhang, Y. Yu and T. Yang, "Identity-Based Provable Data Possession From RSA Assumption for Secure Cloud Storage," in IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 3, pp. 1753-1769, 1 May-June 2022, doi: 10.1109/TDSC.2020.3036641.
[35] Z. Ullah, B. Raza, H. Shah, S. Khan and A. Waheed, "Towards Blockchain-Based Secure Storage and Trusted Data Sharing Scheme for IoT Environment," in IEEE Access, vol. 10, pp. 36978-36994, 2022, doi: 10.1109/ACCESS.2022.3164081.
[36] K. B. Jyothilakshmi, V. Robins, and A. S. Mahesh, "A comparative analysis between hyperledger fabric and ethereum in medical sector: A systematic review," in Sustainable Communication Networks and Application. Singapore: Springer, 2022, pp. 67–86.
[37] G Shrivastava and S. Patel, "Hybrid Confidentiality Framework for Secured Cloud Computing" in 2022 IEEE 3rd Global Conference for

RESEARCH ARTICLE

Advancement in Technology (GCAT), 07-09 October 2022, 10.1109/GCAT55367.2022.9972165.

Authors

Er. Gaurav Shrivastava: He is currently Research Scholar of SAGE University, Indore, He is working as an Assistant Professor in Dept. of Information Technology, SVIIT, Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore, He having more than 12 years of Experienced in Teaching Field, He received his Master of Engineering (Information Technology) with Specialisation in Information Security from Institute of Engineering & Technology, Devi Ahilya Vishwavidyalaya, Indore, and Bachelor of Engineering (CSE) from J.I.T., Borawan, Dist. Khargone, He is Editorial Board Member and Reviewer of many Reputed International Journals; He has published more than 12 research papers in Reputed International Journals of different areas. His current research interests include Information Security, Computer Networks & Cloud Security.

Dr. Sachin Patel: He having more than 19 years of Experienced in Teaching Field. He has Teaching Experience from various reputed Institute (SVCE Indore, ATC Indore, PCST Indore, JIT Borawan and GWPC Khargone) in the Indore region before joining the Sage University Indore. Currently he is Working as an Associate Professor in CSE Department at SAGE University Indore from September 2019. Having B.E. (Computer Engineering) ,M.Tech.(Computer Science) and PHD in Computer Science and Engineering. His research concentrated on the role of "Opinion Mining by using web documents". His areas of interest are Data Mining, Mobile Ad hoc Network, Data Base Management System, and Computer Networking. He has published more than 40 research papers in Scopus and International journals and conferences. He has Published five national patents and 6 Ph.D. scholars are pursuing Ph.D. at Sage University Indore.

How to cite this article:

Gaurav Shrivastava, Sachin Patel, "Secure Storage and Data Sharing Scheme Using Private Blockchain-Based HDFS Data Storage for Cloud Computing", International Journal of Computer Networks and Applications (IJCNA), 10(1), PP: 28-38, 2023, DOI: 10.22247/ijcna/2023/218509.