Original Article

To Use an Ethereum-Based Public Blockchain Network to Provide Confidentiality, Integrity, and Access Control to IoT-Based Medical Healthcare Data

Ochchhav Patel1*, Dr. Hiren Patel2

1*SVKM, KSV, LDRP-ITR, Gandhinagar, Gujarat, India. Email: ochchhavpatel@gmail.com
[2]SVKM, KSV, VS-ITR, Kadi, Gujarat, India. Email: hbpatel1976@gmail.com

***Corresponding Author: -** Ochchhav Patel

*SVKM, KSV, LDRP-ITR, Gandhinagar, Gujarat, India. Email: ochchhavpatel@gmail.com
DOI: 10.47750/pnr.2022.13.S08.562

Abstract

The Internet of Things (IoT) is a network of active and passive objects that interact with each other for information exchange. Due to the immense acceptance of the concept, the IoT utility rate has increased exponentially in recent years. Being an open (and sometimes not-physically-observed) network, the issue of data privacy and security becomes a severe concern on account of the presence of untrusted and malicious users on the network. Most of the existing security approaches are centralized in nature, which leads to bottlenecks and obscurity in public verifiability. Due to its functionality, Blockchain technology could be a solution to these problems, owing to its distributed immutable ledger that is cryptographically secured. In this research, we intend to address the security issues in terms of confidentiality, integrity, and access control mechanisms in the IoT for a healthcare system, where the data files of a particular patient are encrypted and stored on the public blockchain storage in a distributed fashion and accessed through smart contracts. We have observed the efficiency and time complexity of an IoT-based medical health care system with the utilization of Kovan, Binance Smart Chain, Rinkeby, and Matic Blockchain networks. The authors proposed a secure model for IoT-based healthcare systems with the incorporation of the blockchain network.

Keywords: - Ethereum, Healthcare, IPFS, Security.

INTRODUCTION

In the healthcare industry, information technology is becoming increasingly significant and prominent. Healthcare is a vital component of the industry. Patients' body states, such as heart rate, diabetes, electroencephalogram, and other key biomedical signals, can be monitored using various medical monitoring devices and sensors in addition to standard medical examinations for diagnosis or health quality improvement. In the healthcare industry, as well as the rest of the world, data security and privacy are more critical than ever. According to Abdalla [1], the amount of data breaches exposing sensitive healthcare information is increasing. As the volume of healthcare data grows, so do the concerns about data privacy and security. Privacy and security breaches can not only harm your organization's reputation and jeopardize patient relationships, but they can also cost you a lot of money. Personal health information must be stored by healthcare organizations such as hospitals, clinics, and private healthcare agencies. While the majority of healthcare companies ensure that sensitive data is securely stored and encrypted, none have complete control over the security. A single blunder, and your data is vulnerable to a third party.

Cloud computing, augmented reality, artificial intelligence, the Internet of Things, IPFS, and blockchain are all important in medical healthcare. The use of the Internet of Things for real-time patient tracking aids in the delivery of more efficient and effective care. Doctors may access information more easily and give high-quality care well before the patient enters the hospital due to emergency communication via a mobile app. In the healthcare industry, IoT is not a stand-alone solution. To assist healthcare facilities in transforming themselves in a meaningful way, all IoT devices and networks must be coupled with other technologies [2]. Cloud computing is particularly effective in terms of securing data. In healthcare, proper, efficient technologies improve the overall security of patients' electronic medical information, tackle product authenticity and traceability difficulties in the drug supply chain, and allow efficient interoperability across healthcare organizations. Some disadvantages and limitations of cloud storage include usability, vulnerability to attack, bandwidth, accessibility, and data security. Is there any downside to adopting cloud storage in terms of data privacy and security? While cloud computing has made data management easier and offers several advantages in areas like analysis and automation, there are certain risks and cons to be aware of while using cloud storage. The fact that data isn't under your own safe control is one of the most obvious downsides of the cloud storage approach that many firms adopt. Furthermore, the centralized form of cloud servers, as well as the growing number of end access points, raises concerns about certain types of cyber assaults [3]. There are more endpoint vulnerabilities than ever before due to the sheer number of devices that exist today and are continually growing, from smartphones and tablets to the connected home and the Internet of Things (IoT). Phishing and social engineering assaults are becoming more widespread as a result of the open nature of the cloud, which allows you to access core data from nearly anywhere [4].

A blockchain is a public ledger where all the transactions are stored in a chain of blocks. The chain continuously grows when new blocks are joined to it. Blockchain technology has some key characteristics, such as immutability, decentralization, persistency, and anonymity, to solve the abovementioned issues. In blockchain, some core technologies, such as distributed consensus, digital signature-based asymmetric cryptography, and cryptographic hash, enable a decentralized environment. Blockchain provides data integrity because, after a block has been added to the end of the blockchain, it is very difficult to go back and alter the contents of the block. That’s because each block contains its own hash, along with the hash of the previous block. The hash codes are created by a math function that turns digital information into a string of numbers and letters. If that information is edited in any way, the hash code changes as well [5]. To address the issue of reliability, blockchain networks have implemented tests for computers that want to join and add blocks to the chain. The tests, called” consensus mechanisms,” require users to” prove “themselves before they can participate in a blockchain network. Availability of data is important. For example, someone tries to transmit you encrypted and signed data, but the communication connection has been damaged by a government agency. What good is data security if you don’t have the data to begin with? Blockchain provides at least two facets of data security: guarantee of data integrity and availability. Blockchain technology has the potential to revolutionize health care by putting patients at the centre of the healthcare ecosystem and improving health data security, confidentiality, and interoperability [1], [5].

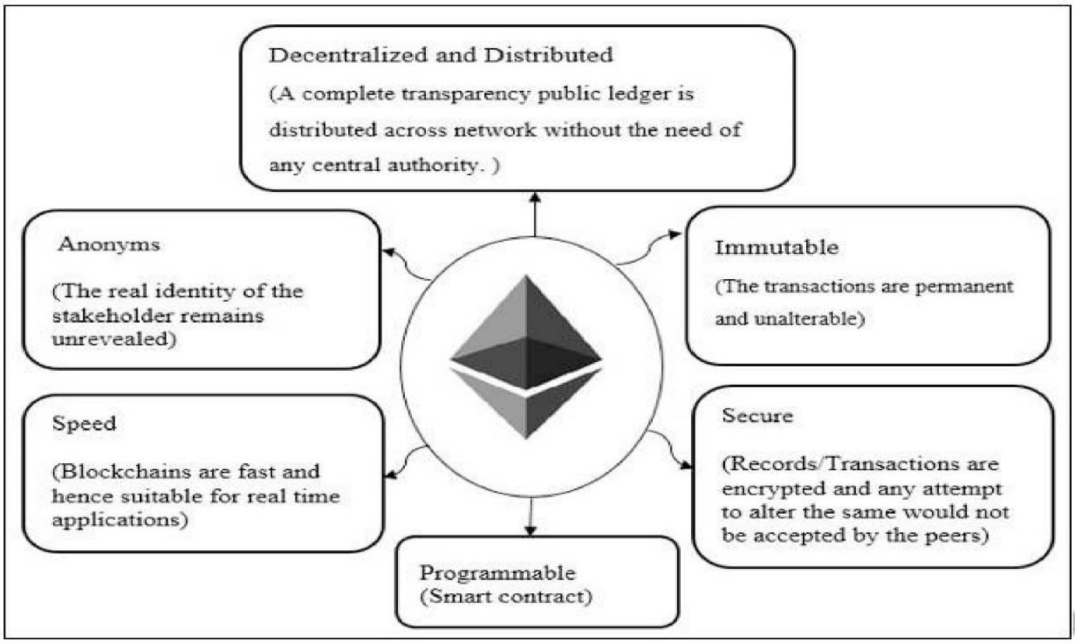

Decentralized and Distributed
(A complete transparency public ledger is distributed across network without the need of any central authority.)
Anonyms
(The real identity of the stakeholder remains unrevealed)
Immutable
(The transactions are permanent and unalterable)
Speed
(Blockchains are fast and hence suitable for real time applications)
Secure
(Records/Transactions are encrypted and any attempt to alter the same would not be accepted by the peers)
Programmable
(Smart contract)

Figure 1: Blockchain Features.

Here Figure 1 illustrates blockchain features. In this research, we look at the healthcare sector, where there are a variety of detectors/sensors that can measure important physical quantities like human temperature, heart rate, oxygen level, and so on. We enhance the security in terms of confidentiality, data integrity, and access control measures in this study. An IoT data is generated in a secure environment and saved on IPFS (Interplanetary File System) storage so that it may be accessed by authorised users over the blockchain network. There are no confidential nodes in the peer-to-peer distributed file system known as IPFS. Local storage is used by IPFS nodes to keep IPFS objects. Each IPFS node connects to the others and shares objects. These objects serve as representations for files and other data structures [6]. In IPFS, a label used to identify content is called a content identifier, or CID. According to that CID, IPFS offers more secure data storage than centralized servers. IPFS is able to offer a unique hash value for each data file that ensures data integrity in our system. Due to IPFS’s ability to store and retrieve data based on CIDs (Content Identifiers), we employed the IPFS platform for data integrity in our implementation. This paper is organized as follows: Section 1 introduces a primer view of healthcare domain, IoT and Blockchain technology overview, data security issues, and how to address security issues using Blockchain. The work that has already been done using supporting technologies for the medical healthcare system, like IoT and Blockchain, is covered in Section 2. Research proposal, workflow, and IoT-Blockchain-based medical healthcare system are all described in Section 3. In Section 4, multiple blockchain networks are used to discuss the latency of the Blockchain network. The implementation of our suggested work with mathematical derivation is covered in Section 5. In section 6, we outline results and numerous security risks that we have implemented. Section 7 describes the conclusion of our examined work. Section 8, which is the final section, is a future direction for an innovative researcher.

RELATED WORK

Different architectures and algorithms have been proposed by various researchers to provide authentication for accessing

IoT data in recent times. In this section of this paper, we are going to study a few of them. Alam, T. [1] has proposed mHealth, which is a blockchain-based predicated mobile health fortified system that intends to provide a unique identity to every connected contrivance in a network to offer data storage and sharing with security and transparency. A single-level authentication mechanism (e.g., password) is incorporated in IoT-based systems, which makes them vulnerable to various types of attacks. A utilizer-and contrivance-based authentication-predicated technique has been proposed by R. S. Joshitta [2]. In this mechanism, all the devices that have been connected to the network are identified by a unique RFID number. The users have been bifurcated among two categories, viz., privileged users and ordinary users. Patients' continuously streaming medical data is stored on the cloud, while their meta-data, such as the CCR (continuity of care record), is stored on the gateway server. The concerned users interact with the gateway server through their credentials.

Dwivedi et al [3] offer secure management and examination of healthcare data through blockchain. They have incorporated security and privacy properties into the proposed model to truncate sundry attacks such as denial of service and modification attacks. The Internet of Medical Things (IoMT) [4] requires enormous infrastructure for storage and processing of a substantial volume of medical data. The need for IoMT applications and platforms on a centralized cloud is cooperating with security, so an IoMT system with a blockchain data structure is claimed to provide better security and privacy than a central data storage repository. In the IoMT system, a large amount of data is stored on a decentralized storage platform, whereas the hash codes are on the blockchain. A reliable and secure healthcare architecture has been proposed by the authors [5] using the ECC algorithm over the CoAP protocol. The proposed approach provides an effective authentication mechanism with high security.

Griggs et al. [6] proposed blockchain-based smart contracts to enable secure analysis and management of medical sensors in the healthcare system. They have used a private blockchain, based on the Ethereum protocol, where the sensors communicate with IoT devices, which in turn call smart contracts and write records of all actions on the blockchain. In [7], researchers have mentioned the Oracle as a smart device, which communicates directly to the smart contracts and that could assess the provided data and issue alerts to the patient as well as to the healthcare provider. One of the modules deals with the process of fetching and sensing the data from the wearable contrivances and the biosensors that are either worn by the patients or that are present in the environment in which patients are monitored. In [8] S. Chakraborty, introduced a methodology that overcomes the security and privacy issues of the data generated by the conventional healthcare system. The new abstract model based on blockchain and IoT could offer solutions for numerous problems. Each patient or doctor could access their data via a dashboard connected to the blockchain server using the REST API [9]. The patient's record history could be stored on the distributed storage [10], which can be linked to the blockchain with a hash value.

In order to address the security concerns in the current methodologies for exchanging healthcare data, Shen and his team introduced consortium based Med Chain [11] data-sharing scheme, which integrates structured P2P network techniques, digest chains, and blockchain technology. A session-based healthcare data-sharing system that is based on Med Chain has been developed, bringing flexibility to data sharing. A secure electronic health record (EHR) system based on attribute-based cryptosystems and blockchain technology has been proposed by researchers [12] to achieve confidentiality, authentication, integrity, and support for fine-grained access control of medical data. They have implemented digital signatures using identity-based signatures (IBS) and have used attribute-based encryption (ABE) and identity-based encryption (IBE) to encrypt medical data. Bloc HIE is a blockchain-based network for exchanging medical data that is suggested by researchers in [13]. They looked into the various specifications for exchanging healthcare data from various sources. In order to meet the demands of both privacy and authenticity, they have employed two loosely coupled blockchains to manage various types of healthcare data and combined off-chain storage and on-chain verification. Blockchain technology is an evolving and upcoming technology that enables data sharing in a transactional and decentralized fashion. In the healthcare domain, blockchain can provide a balance between privacy and access to electronic health records.

The primary goal of Health Block [14] is to employ blockchain technology to increase the security and privacy of electronic health records (EHRs). They created an innovative system that takes advantage of decentralized databases to get past issues with centralized storage. EHRs for patients are kept in the decentralized Orbit DB with the Interplanetary File System database (IPFS). Additionally, they have established a blockchain network based on the Hyperledger fabric by using Hyperledger Composer to record hashes of saved data and control access when retrieving it. The proposed blockchain-based architecture aims to overcome known security issues in current systems for smart healthcare and strengthen the stability of healthcare management systems. The Smart Med Chain architecture, an end-to-end blockchain-baseband privacy-preserving solution, was created by researchers [15] for data exchange in the s-healthcare context. Encrypted health data has been stored using Hyperledger Fabric and the Inter Planetary File System (IPFS), a distributed data storage solution with exceptional durability and scalability. Scalable machine learning applications like healthcare use federated learning technologies.

Health data are encrypted to conduct fine-grained access control in the Health chain system, which is developed by researchers in [16]. It is based on blockchain technology and is designed to protect the privacy of vast amounts of health data. In order to prevent medical conflicts, Health chain ensures that both IoT data and medical diagnoses cannot be altered or removed. In the mentioned architecture, they have used the user chain (public blockchain) network with proof of work consensus and the Doc chain network with Practical Byzantine Fault Tolerance (PBFT) consensus. In Health chain, the IPFS system is managed and maintained by a consortium of healthcare providers.

Researchers described the BC Health architecture in [17], which addresses the issue of transparency and access control being compromised, and which lets data owners to specify the appropriate access controls over their privacy-sensitive healthcare data. For the purposes of storing access policies and data transactions, BC Health is made up of two distinct chains. To boost system performance and scalability, they employed a new, modified blockchain network and the Proof-of-Authority (PoA) consensus process. They used Python to implement BC Health's components and measure how well it worked. The COVID-19Pandamics system in [18] is based on a peer-to-peer network powered by the distributed Interplanetary File System paired with on-chain tagging, as well as on the use of cryptographic generating techniques to enable a secure manner of sharing medical data. Using an open-source version of the Pretty Good Privacy (PGP) encryption technique, medical data is safely protected. The suggested architecture encrypts medical data using asymmetric cryptography using the recipient's public key. Before the data is re-pushed for storage on a peer-to-peer file storage system run by an IPFS cluster, encryption takes place at the client side. The system uses public-key cryptography to manage each participant's identify information. Identity is pseudonymous to protect privacy; nonetheless, the smart contract can link users' social security numbers or any other widely used form of identification with their Ethereum public addresses. The secp256k1Elliptic Curve Digital Signature Algorithm is used to create an Ethereum public/private key pair. On the Ethereum blockchain, this pair serves as the authentication mechanism. Singh and his team [19] proposed architecture that is supported by federated learning and blockchain for the purpose of protecting privacy in smart healthcare, where blockchain-based IoT cloud platforms are employed for data protection. The Ethereum blockchain, Hyperledger Fabric, cryptographic techniques, and IPFS were some of the technologies that researchers in [20] explored in order to address EHR (Electronic Health Records) related issues and various technological solutions. To solve the data security and privacy issues with EHRs, they have created HER Chain. The ability to decentralize patient data via a consortium blockchain and IPFS for distributed data storage is made possible by dual blockchains built on Hyperledger Sawtooth.

In order to ensure privacy and security in IoT-driven smart cities, researchers presented the Privacy-Preserving and Secure Framework (PPSF) [21] architecture, which is an intelligent blockchain framework that combines blockchain with machine learning algorithms. A two-level privacy system and an intrusion detection technique are the two primary mechanisms on which the proposed PPSF is built. First, a blockchain module is created in a two-level privacy scheme to securely transmit the IoT data, and the principal component analysis (PCA) method is used to reshape the original IoT data. Gradient Boosting Anomaly Detector (GBAD) is used in the intrusion detection technique for training and assessing the proposed two-level privacy strategy based on two IoT network datasets, ToN-IoT and BoT-IoT. They also suggest a fog-cloud architecture that integrates blockchain and IPFS for use in deploying the PPSF framework. Researchers introduce "Privy Sharing", [22] a novel framework built on blockchain technology for safe and secure IoT data exchange in a smart city setting. The blockchain network is divided into different channels that each process a particular sort of data, such as health, smart auto, smart energy, or financial information, and each comprise a small number of authorised organisations. Additionally, the smart contracts contain access control rules that regulate who has access to the data of users within a channel. They set up the Privy Sharing business network model on Hyperledger Fabric version 1.4 and introduced a method of compensation for users who shared their data with third parties in the form of a virtual currency called "Privy Coin."

A two-level privacy engine and an anomaly detection engine based on deep learning are the two core engines of the proposed SP2F [23] framework. To authenticate data and prevent data poisoning attacks, the two-level privacy engine uses an enhanced proof of work (ePoW) system based on smart contracts and the blockchain. Using a sparse autoencoder (SAE), data is changed into a new encoded format to protect against inference attacks. Two publicly available IoT-based datasets are utilised to train and test the proposed two-level privacy engine's performance in the anomaly detection engine, which employs a stacked long-short-term memory (SLSTM). The possibility of incorporating permissioned blockchain technology into the current procedures of the coffee supply chain [24] has been studied by researchers. In the context of the coffee supply chain business, the issues with data integrity, provenance transparency, privacy, and security are highlighted more specifically while also aiming to generalise the solution to effectively handle other supply chain operations. This study focuses on identifying permissioned blockchain platforms to aid supply chain industry stakeholders in using a less corruptible substitute for traditional web technology and to enable a more positively nuanced blockchain system that strikes the best balance between traditional web technology and a public blockchain, including privacy protection and security.

According to our review of the literature, the majority of academics have employed a private blockchain (the Hyperledger network) to accomplish data security; by contrast, we have used a public blockchain network to secure data. In the proposed system, anyone can join the system due to the public blockchain platform. We use the public key of the receiver to send the data file, the patient's identity remains anonymous, and only the sender's public key is displayed on the receiver's side. When compared to the permissionless (public) blockchain, the permissioned (private) blockchain is less secure because there are very few validators who check the blocks or transactions. A permissionless blockchain requires that each block and transaction be verified before being included. Therefore, we prefer the mentioned sources in an IoT-based healthcare system using IPFS storage on an Ethereum-based public blockchain.

OUR PROPOSAL

An IoT device can be used in a variety of applications, ranging from smart cities and homes to healthcare and agriculture to education, since they are flexible not only in terms of deployment and management but also in terms of interfacing with other networks or devices. In this research, we consider the healthcare domain, where various kinds of detectors/sensors are available that estimate significant physical quantities such as human temperature, pulse rate, oxygen level, etc. There are various layers, which are important from the IoT's functionality point of view. The perception layer of the IoT is responsible for ordering data and transmitting it to the network layer. It also enables one gadget to collaborate with other gadgets. The network layer is accountable for managing the communication and transmitting the piled-up data from the perception layer to the storage servers via gateways. The application layer manages the gathered data, and processed information is sent to the applications or end users' community.

Here, a data file is encrypted using the AES-CBC (Advanced Encryption Standard-Cipher Block Chaining) cryptographic algorithm, and that encrypted file is stored on IPFS storage. The robustness of AES, a symmetric cryptosystem, is confirmed by the theory of computational complexity. It is related to a procedure that explains how to encrypt the data blocks. There are numerous operational modes. We operate in CBC mode, which ensures the confidentiality and integrity of the data. As a more secure alternative to the electronic codebook (ECB), cipher block chaining is a popular technique for encrypting lengthy communications. The main advantage of the AES algorithm is the availability of multiple key lengths. The size of the key used to secure the communication has a direct impact on how long it takes to decrypt data using a given encryption algorithm. Due to the blockchain-based mechanism, only authenticated users can access the patient's data file from the data storage.

Proposed Model

In the proposed system medical data are generated by various medical sensors. Generated file is protected by AES symmetric key. Protected data file is stored on IPFS platform. In our implementation we have used IPFS storage because of it support content-addressed storage and uses a decentralised distributed filesystem. The widely used HTTP system of today is in marked contrast to the IPFS protocol. A centralized, address-based protocol is HTTP. Users retrieve resources using HTTP entirely based on where they are on centralised servers. This system has a number of drawbacks. The vulnerability of a single device increases when resources are concentrated on it. Over HTTP, a distributed file system that is content-addressed, like IPFS, has a number of advantages. These benefits can be attained in an IoT setting by first creating the infrastructure to support IPFS. In a cryptographic process, AES key is encrypted by receiver's public key and decryption of that key is done by private key of receiver that process is done by smart contract. As per the analysis of previous work, majority work is done by permissioned blockchain or consortium blockchain. When compared to permissionless blockchain, permissioned blockchain is less secure because there aren't any or very few validators who check the blocks or transactions. A permissionless blockchain requires that each block and transaction be verified before being included. Therefore, we prefer the mentioned sources in IoT based healthcare system using IPFS storage, on ethereum based public blockchain. The data block in this instance, according to Figure 2 has the block's current hash value, its prior hash, and a data value that contains the hash of the patient data file (D'Patient) and K'S.

Healthcare applications correspond with the layered architecture of the IoT. The first level includes sensors or medical gadgets, and works as a unit for data procurement such as pulse rate, oximeter, and human body temperature. The second level incorporates communication and the services that collect data from the first layer and deliver it to the next layer. The third layer is for data processing and accepting results. Medical researchers store the patient's data for clinical as well as research objectives, so the healthcare system should demand more cautiousness about data because it concerns the life of the patient. The users are divided into mainly two categories: primary users, including doctors, nurses, and nearby relatives; and secondary users, including medical insurance companies, researchers, and drug inventors. As IoT works on top of the traditional Internet, it inherits the existing underlying vulnerabilities. Hence, (lightweight) cryptographic measures need to be taken for issues such as confidentiality, integrity, and authentication for resource-scarce IoT.

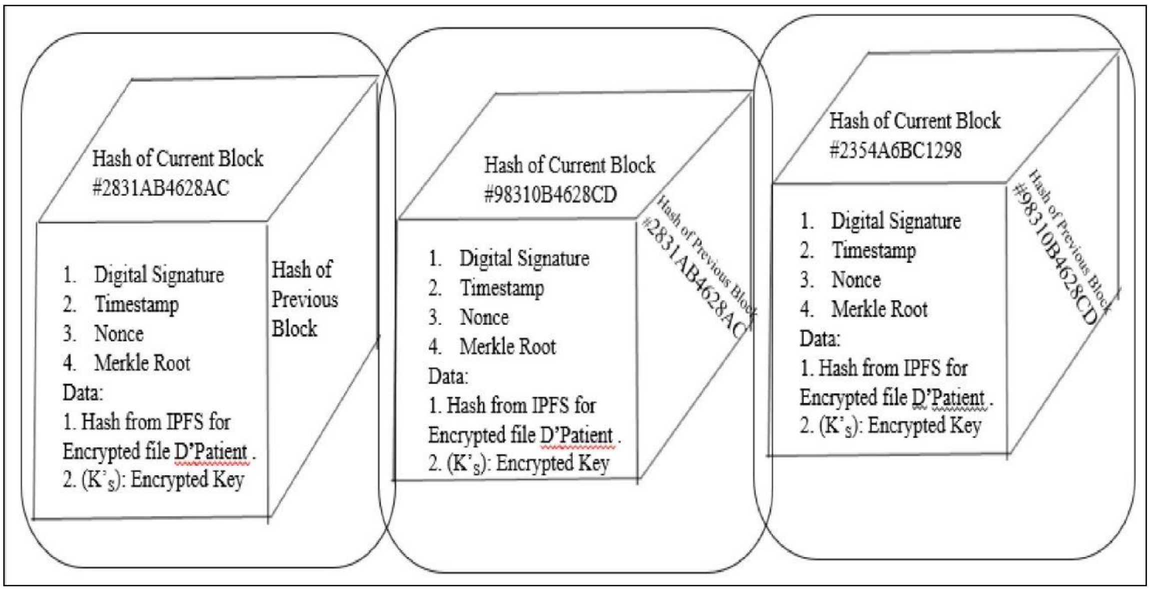


Figure 2: Structure of the block in a Blockchain.

Proposed Work

The Internet of Things suffers from weak connectivity, deprived scalability, a lack of trust, and destitute security. Other concerns are node addressing, node identification, device heterogeneity, and energy constraints [25]. Generated data from the data progenitor, such as the pulse rate, the oxygen level, the temperature, etc., is stored in a file. The data is saved using a specific file naming convention provided by the RFID device. The data file for the patients will be generated in the .json file format. Patients and medical systems are breached by hackers from outside the healthcare institution in order to steal and acquire data, mostly for financial benefit. For instance, they could make false claims for health insurance using patient personal information. Hackers that demand a ransom from healthcare institutions in exchange for recovering patient data systems are another form of external theft. Curiosity (unwarranted access to data not related to the delivery of care) is another problem with healthcare data security. The remaining instances of insider abuse are unintentional behaviours like human mistakes, such as inputting incorrect information into healthcare data or clicking on a phishing email. Table 1 shows different types of attacks.

Table 1: Type of Attacks.

Attacks ↓	**Confidentiality**	**Integrity**	**Availability**
Snooping	**Yes**	--	--
Traffic Analysis	**Yes**	--	--
Modification	--	**Yes**	--
Masquerading	--	**Yes**	--
Replying	--	**Yes**	--
Repudiation	--	**Yes**	--
Denial of Service	--	--	**Yes**
Eavesdropping	**Yes**	--	--
Message Tempering	--	**Yes**	--
Brute Force	**Yes**	--	--

Table 2: Acronyms.

Notation	**Description**
KS	Shared Secret Symmetric Key
K′S	Encrypted Symmetric Key
DPatient	Patient’s Data
D′P	Patient’s Encrypted Data
EKS	Encryption using key KS
DKS	Decryption using key KS
IPFS	InterPlanetary File System
CCR	Continuity of Care Record
HD	Hash of Data (D′Patient)
PUR	Public Key of Receiver
PRR	Private Key of Receiver
PBFT	Practical Byzantine Fault Tolerance
PGP	Pretty Good Privacy

Algorithm 1: Data Encryption

```
1: Function Encryption: Input Data File (DPatient), Output Encrypted File (D’Patient)
2: Collect a symmetric key KS (from Key Distribution Centre)
3: Select the encryption algorithm (E.g., AES)
4: Generate a random initial vector IV
5: D’Patient ←EKS (DPatient, IV)
6: Return the encrypted file D’Patient on public channel and IV & KS on private channel
```

The way that digital signatures operate is by demonstrating that a document has not been changed, either purposefully or unlawfully, since it has signed. This is accomplished through digital signatures, which use the sender’s private key to encrypt a distinct hash of the message or document. The hash created is specific to the message or document, and if any part of it is changed, the hash will be completely altered. The message or digital document is then signed digitally and transmitted to the receiver. Then, using the sender’s public key, the recipient creates their own hash of the message or digital document and decrypts the sender’s hash, which was integrated into the original message. The message or digital document is hashed by the recipient and compared to the sender’s decrypted hash; if they match, the message or digital document has not been altered and the sender is verified. Based on the implementation view, patient data file is encrypted with the symmetric key of the AES encryption technique and that key is encrypted with the recipient’s public key. The receiver’s keys are kept in the database when he visits that platform to retrieve a data file, at which point he use his private key to unlock the original data file. In the encryption algorithm, we encrypt the data file (DPatient) by using the symmetric key (KS) and initial vector IV. As per the mentioned in algorithm 1, After the use of symmetric key and initial vector (IV), it will generate ciphertext (D’Patient). After data file encryption, that symmetric key (KS) will be encrypted by the receiver’s public key (PUR) and stored on the blockchain platform for further purposes. The symmetric key (KS) for the original data file (DPatient) has been decrypted using the receiver's private key (PRR), as shown in Algorithm 2. The encrypted file (D'Patient) with a unique id, will be stored on IPFS storage. The acknowledgement of the submitted file will be received from the IPFS in the form of a hash value. IPFS storage is also more public, so confidentiality of data is required. This could be a violation of certain types of data storage, which expose private information. IPFS can provide proof of authenticity to the owner of that content. It seems IPFS distributes expeditious and secure fault tolerant file storage for content. As IPFS evolves, it could utilize a privacy layer that can hide personal data that is withal encrypted at rest, so there would be no contravention of exposing anything confidential.

Algorithm 2: Data Decryption on Receiver side

```
Input: Encrypted File (D'Patient), EKS :: Key Encryption using receiver's public Key (PUR)
Output: Decrypted Data File (DPatient)
Function Decryption ((D'Patient), (EKS ),KS)
(KS)←−DecryptionAsym((EKS ),PRR)
DPatient←−DecryptionSym((D'Patient),KS)
End Function
```

The data file (DPatient) is encrypted using an AES symmetric key (KS). Here, the receiver's public key (PUR) of the RSA algorithm, is used to encrypt the symmetric key (KS) and generate the encrypted key K′S. The encrypted symmetric key K′S will be stored on the blockchain platform. Algorithm 3 demonstrates how to deliver a patient's data file (DPatient) to IPFS storage and compute the hash value (HD) for that file. The retrieved file from the IPFS will be in encrypted format, so it would be required to decrypt the file (D'Patient) to convert it into the original data file (DPatient). Whenever end users such as doctors, nurses, and relatives want to access the data file of a particular patient, they will be required to pass an authentication process. The authentication mechanism of the user succeeds the security mechanism. Here, the smart contract will find the hash value of the requested file from the blockchain and send the response in the form of the hash value of that file to the primary validated user. The verified user will use the hash value to access the IPFS storage and return with the desired file when the hash value matches.

By using cryptographic hashes to sign the encrypted patient data file, the integrity of the shared data is protected. A smart contract is then used to store the results of the hashing process on the blockchain. Using references to the IPFS content identifiers, a smart contract controls access to the encrypted patient records. Additionally, the smart contract's current state implements logic for access control to protect the permissions for information access. The data owner is the owner of the smart contract and has the authority to allow access to the medical data as well as to write certain metadata to the blockchain. A list of data pointers and related permissions for the patient data file are specified in the contract. A reference to an IPFS content identifier is used at this layer to signify access to the medical data file. Based on the intended recipient's public key that is stored on the blockchain, indexing to the data is done.

Algorithm 3: Interplanetary File System for Distributed Storage

```
Function IPFS
Patient's Data File (D′Patient) send to the IPFS for Data Storage
Input Parameters: (IPFS, (D′Patient))
Output : (HD)
Step 1 - Data Storage Request
Source Node (SN) sends Request to IPFS Storage
IF
Ethereum address of source node exists in blockchain
IPFS divides the data into small chunks
Hash of Data file (HD) is calculated
Step 2 - Data Acquirement Request
SN Requests the data from IPFS using (HD)
IF
(HD) is present on IPFS server
then
IPFS aggregates the data
IPFS sends Protected Data File (D′Patient) to the SN
ELSE
Message (The data is not present)
End IF
ELSE
SN is a malicious node
End IF
End Function
```

BLOCKCHAIN NETWORK LATENCY

Blockchain technology presents new tools for authentication and authorization in the digital world that prevent the demand for a kind of unified administrator [26]. As an outcome, it enables the creation of new digital relationships. Blockchain technology uses a smart contract protocol or rule that is immutable, which means once it is deployed it stays in place indefinitely. In the event that a revised version of the present contract is created due to a production bug, then the manual transfer of stored data is required, which is an awkward process. In blockchain, performance measurement is challenging as it is very complex to duplicate a creation-like environment for performance testing. It needs to be tested for network latency based on block size, network type, expected transaction size, and how long a query takes to return results with the technical authentication protocol.

We have put the transactions into practice on multiple test blockchains and calculated the amount of gas needed to establish smart contracts on the blockchain network. Additionally, we calculated the amount of gas needed to upload the hash of the encrypted AES key and protected data files to the blockchain network. For transactions, we have used the blockchain networks of Kovan, Rinkeby, BSC, and Matic.

The comparison of various blockchain networks for the amount of gas used during the deployment of smart contracts on the blockchain network is shown in Figure 3. We draw a conclusion and observe that the MATIC network uses less gas to deploy smart contracts. In terms of gas requirement comparisons on Ethereum, the Kovan and Rinkeby test networks used more gas than the Binance and Matic networks. Depending on the network and other factors, your transaction will execute significantly more slowly if you pay a smaller fee.

IMPLEMENTATION

There are two stages to our experimental work. An RFID reader, tags, medical sensors, and a Raspberry Pi was employed in the first step of IoT deployment. The Ethereum platform, IPFS for distributed storage, Solidity as the programming language, Metamask (a blockchain wallet), and other tools were used to create the blockchain side in the second phase. In our successful execution, we combined two technologies and explained how medical data is created and protected in the Internet of Things context. Additionally, it detailed how to save that data in IPFS storage and how the blockchain network might access it. The patient's unique ID was generated in a file by the RFID device. The resulting data is filtered to meet specifications, and the file is then encrypted using cryptography. Using its built-in Python library, the Raspberry Pi can communicate with a range of sensors and RIFD tags. The Berryboot operating system, which serves as a basis for installing any Raspbian operating system and can function as an all-purpose operating system, was used to install the Raspbian operating system. A Python package called MFRC522.py, which is used to read and write data from and to the RFID tags, is also used to integrate the Raspberry Pi with the RFID RC522 chip.

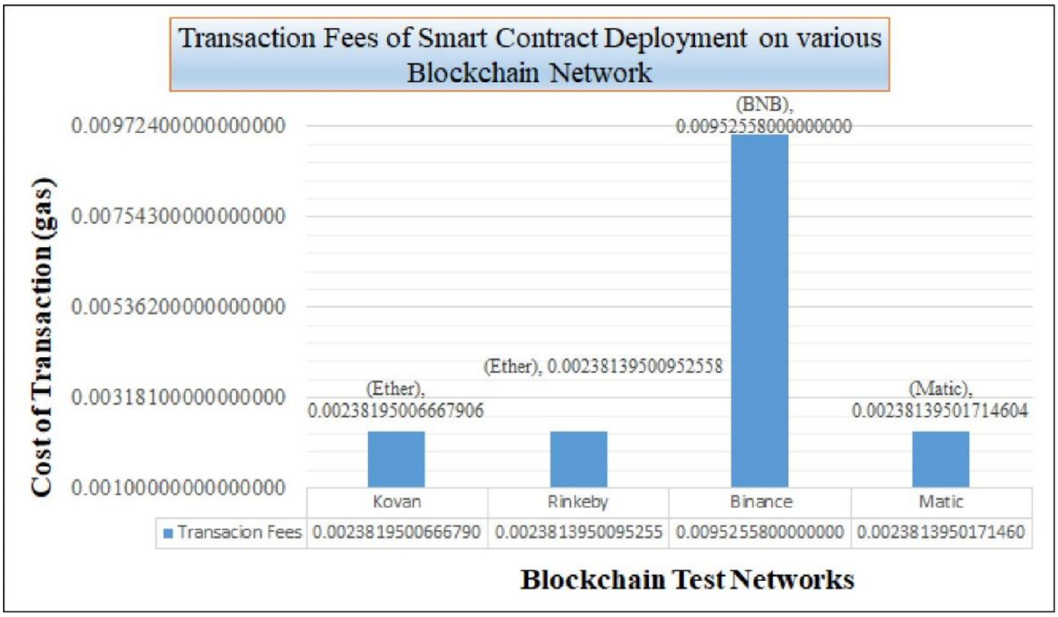


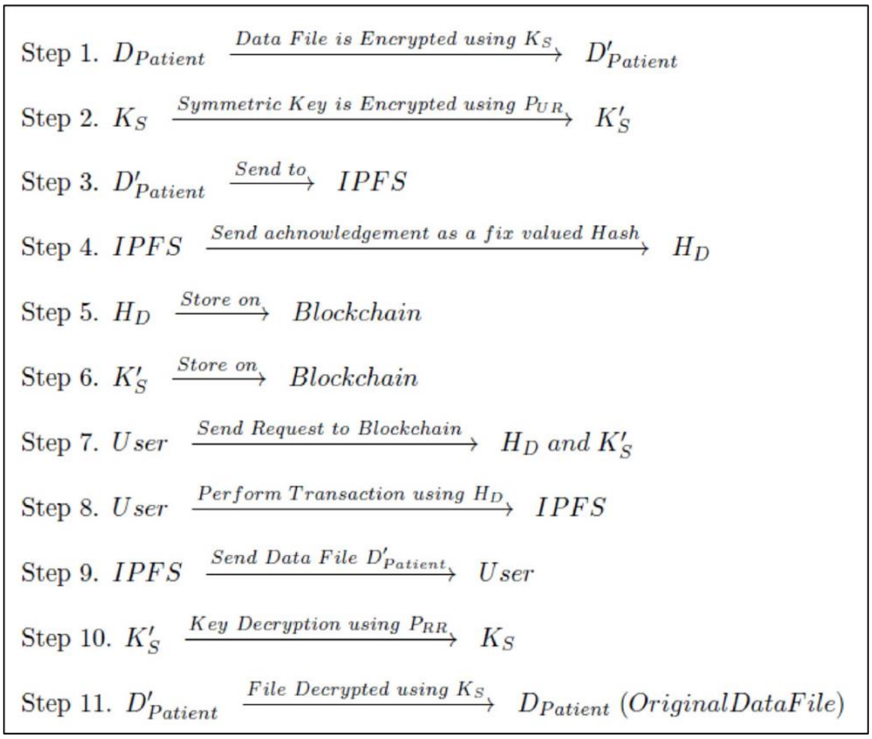
Figure 3: Comparisons of blockchain networks with transaction fees for smart contract deployment.

IPFS storage generates the hash value of a particular patient data file and sends it back to the IoT platform. First, we encrypt that data file on the IoT platform using the Python objcrypt library in the form of an AES-CBC encryption method, where the block size in the AES-CBC algorithm is 128 bits. We stored encrypted content into a file named the same as the ID of the RFID tag, then uploaded the encrypted file to the IPFS storage. IPFS storage returns a fixed-length hash value (HD) of the stored file. This generated hash value (HD) is sent to the blockchain network for future reference.

Mathematical Derivation of Implementation

Here, the patient data file (DPatient) is encrypted using a symmetric key KS. The encrypted data file (D′Patient) is sent to

IPFS storage and, as an acknowledgement, IPFS returns the hash value (HD) for the stored file. There are two values stored on the blockchain. One of them is the hash value (HD) of the protected file (D′Patient) and the other is KS. The symmetric key (KS) is encrypted using the public key PUR of the receiver. Here is an encrypted symmetric key K′S stored on the blockchain network. If anyone wants to access a data file (DPatient) then they have to come on the blockchain platform and use the HD and K′S. Using the hash value (HD) of the data file, the receiver can retrieve the data file in protected mode due to the file encryption method. The receiver can decrypt that file using a symmetric key (KS) but that key is also encrypted using the receiver’s public key. The key KS is decrypted using the receiver’s private key (PRR) and, using that key, the protected file (D′Patient) is decrypted and converted into the original data file (DPatient). The following steps (Figure 4) indicate the life cycle of the entire process of sending and receiving data files (DPatient) from data origin to data consumer.



Step 1. $D_{Patient} \xrightarrow{Data\ File\ is\ Encrypted\ using\ K_S} D'_{Patient}$

Step 2. $K_S \xrightarrow{Symmetric\ Key\ is\ Encrypted\ using\ P_{UR}} K'_S$

Step 3. $D'_{Patient} \xrightarrow{Send\ to} IPFS$

Step 4. $IPFS \xrightarrow{Send\ achnowledgement\ as\ a\ fix\ valued\ Hash} H_D$

Step 5. $H_D \xrightarrow{Store\ on} Blockchain$

Step 6. $K'_S \xrightarrow{Store\ on} Blockchain$

Step 7. $User \xrightarrow{Send\ Request\ to\ Blockchain} H_D\ and\ K'_S$

Step 8. $User \xrightarrow{Perform\ Transaction\ using\ H_D} IPFS$

Step 9. $IPFS \xrightarrow{Send\ Data\ File\ D'_{Patient}} User$

Step 10. $K'_S \xrightarrow{Key\ Decryption\ using\ P_{RR}} K_S$

Step 11. $D'_{Patient} \xrightarrow{File\ Decrypted\ using\ K_S} D_{Patient}\ (OriginalDataFile)$

Figure 4: Mathematical derivation steps for file encryption and decryption process.

Model Implementation

In our implementation, we have used smart contracts, which are written in the Ethereum Solidity programming language, for IoT-based healthcare systems to achieve data confidentiality, integrity, and access control mechanisms. All the medical sensors have been attached to the patient’s body, and the generated data is stored in a unique file that is engendered by the RFID device. Figure 5 shows the cryptographic process where we have used a blockchain-based authenticated mechanism to access the patient’s data file from the distributed storage. The data file is encrypted using the cryptographic technique AES-CBC, and that encrypted file is kept on IPFS storage. In the proposed system, patients can send the data file to doctors, insurance companies, other hospitals, and relatives using the Ethereum wallet address of the recipient.

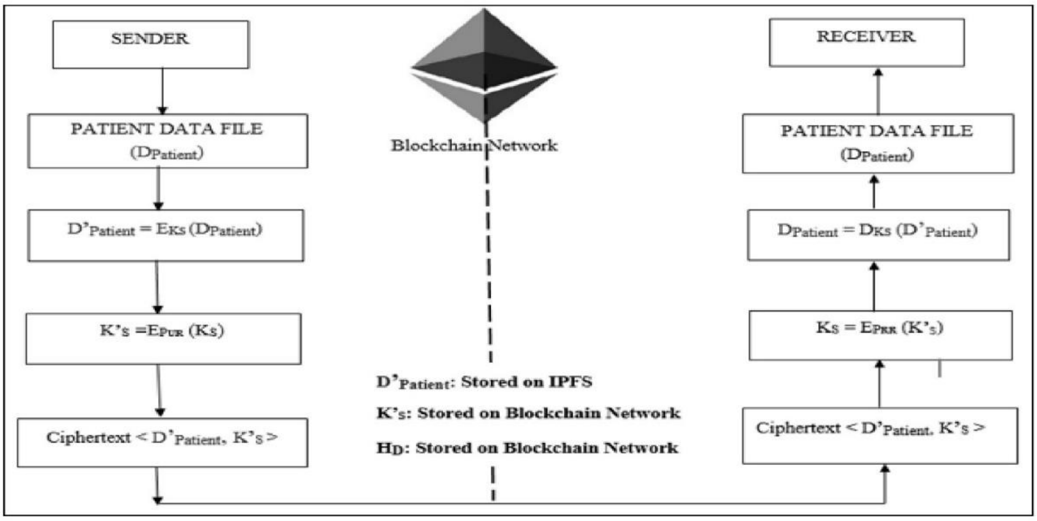

SENDER
PATIENT DATA FILE (DPatient)
D’Patient = EKS (DPatient)
K’S =EPUR (KS)
Ciphertext < D’Patient, K’S >
Blockchain Network
D’Patient: Stored on IPFS
K’S: Stored on Blockchain Network
HD: Stored on Blockchain Network
RECEIVER
PATIENT DATA FILE (DPatient)
DPatient = DKS (D’Patient)
KS = EPRR (K’S)
Ciphertext < D’Patient, K’S >

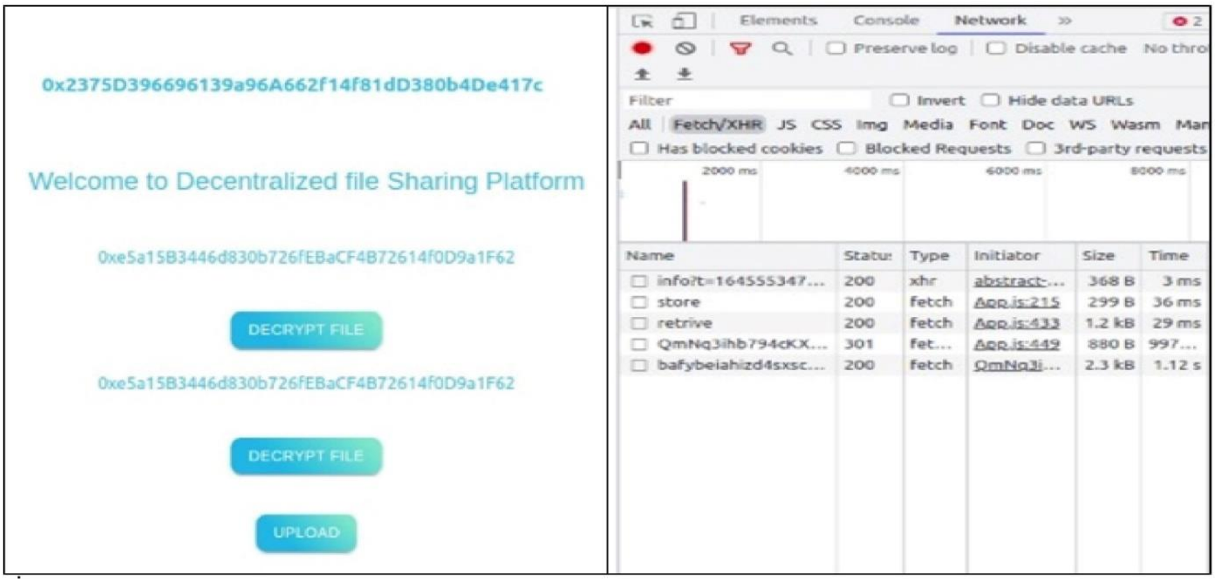
Figure 5: Cryptographic process of proposed model.

The patient sends a data file to the recipient using their Ethereum wallet address, and the recipient must come to the blockchain platform to retrieve the data file. At that time, the public key and private key of the receiver are stored in the database. If patient want to send data file to particular entity (doctor) then he has to use of doctor's public key first. The fact that public blockchains can act as the foundation for almost any decentralized solution makes them incredibly important. A secured public blockchain is additionally protected from data breaches, hacker attempts, and other cyber security risks by the large number of network participants joining it. A blockchain is more secure the more users it has.

In private blockchain, through an invitation that includes authentication and verification of their identity and other necessary information can users join a network. The validation is carried out either by the network operator(s) or by a precisely specified protocol that the network has developed using smart contracts or other automated approval techniques. The data file is encrypted with the receiver's public key and decrypted with the receiver's private key; this process is handled by smart contracts after successful blockchain execution. A list of sender addresses with data file is available on the receiver side, which is mapped with an IPFS hash value. Only authenticated users may access the patient's data file from the data storage as a result of the blockchain-based method, avoiding any form of network assault.

File Sending and Receiving via Blockchain Network

The data file (DPatient) is stored on the IPFS platform and is protected using a cryptographic algorithm (AES). IPFS storage returns the hash value (HD) for that stored file. The retrieved hash value and symmetric key are stored on the blockchain platform. The symmetric key (KS) is encrypted by the receiver's public key (PUR). If the recipient wants to obtain the original data file (DPatient) from IPFS, they must go to the blockchain platform and use their private key (PRR) and the hash value (HD) of the stored data file to decrypt it, as illustrated in Figure 6.



0x2375D396696139a96A662f14f81dD380b4De417c

Welcome to Decentralized file Sharing Platform

0xe5a15B3446d830b726fEBaCF4B72614f0D9a1F62

DECRYPT FILE

0xe5a15B3446d830b726fEBaCF4B72614f0D9a1F62

DECRYPT FILE

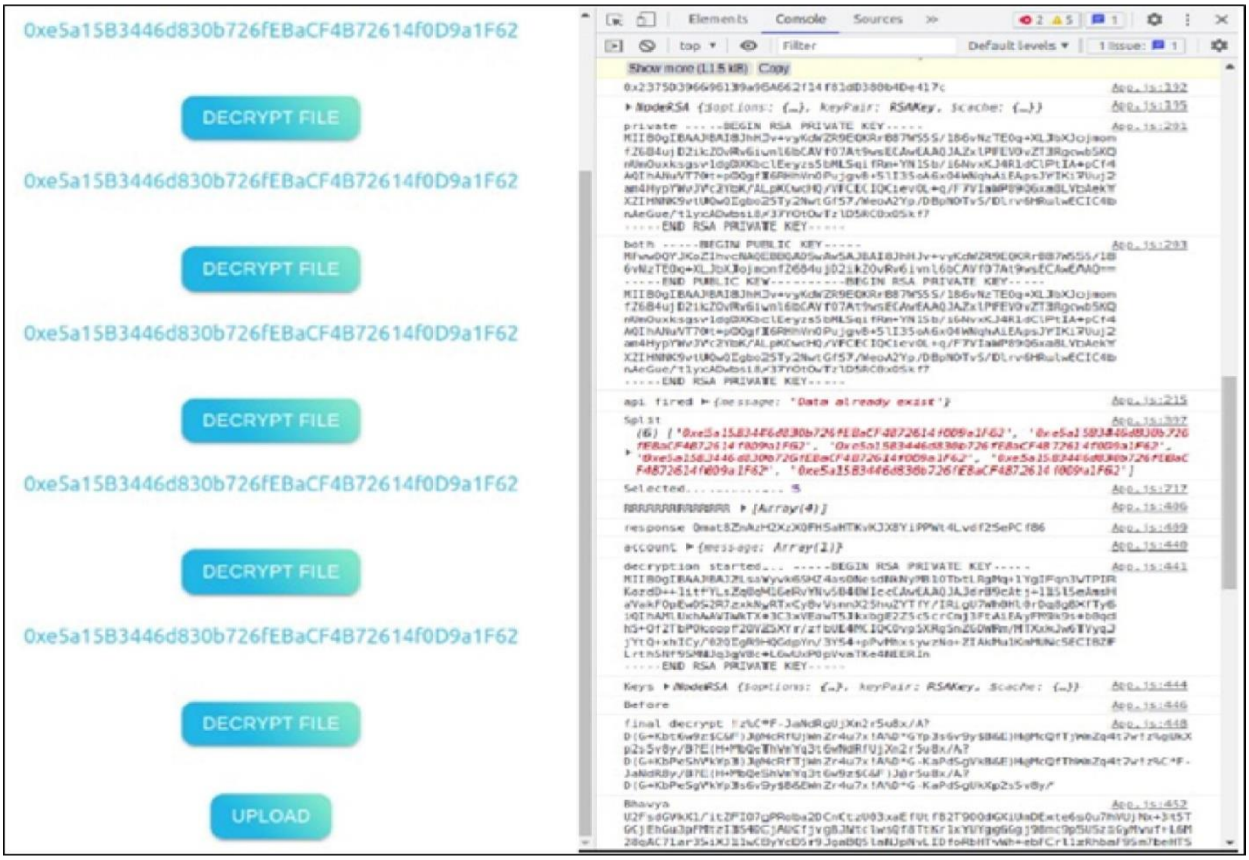
UPLOAD

Name	Statu:	Type	Initiator	Size	Time
□ info?t=164555347...	200	xhr	abstract-...	368 B	3 ms
□ store	200	fetch	App.js:215	299 B	36 ms
□ retrive	200	fetch	App.js:433	1.2 kB	29 ms
□ QmNq3ihb794cKX...	301	fet...	App.js:449	880 B	997...
□ bafybeiahizd4sxsc...	200	fetch	QmNq3i...	2.3 kB	1.12 s

Figure 6: Platform that shows the received data file with the sender's address.

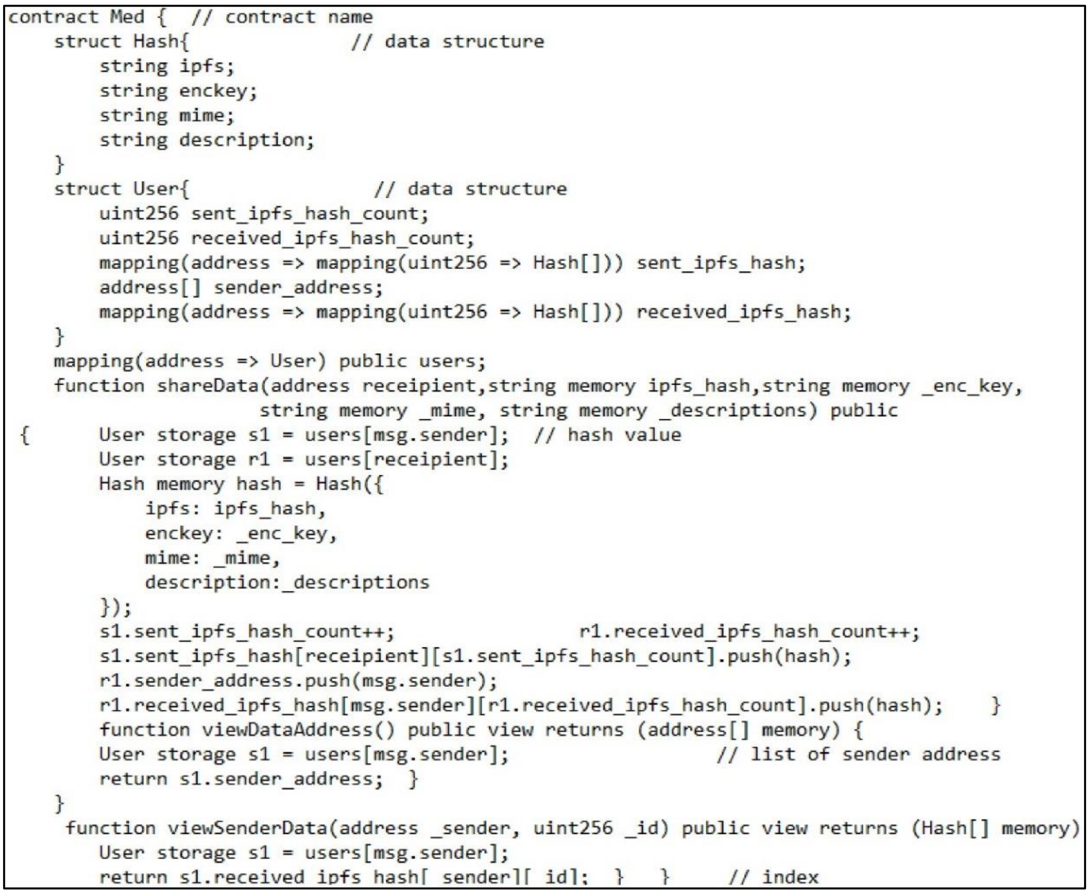
As we know, private and public keys are both halves of an RSA (Rivest-Shamir-Adleman) key pair. The RSA public key is used to validate digital signatures, whereas the RSA private key is used to create them. DES or AES data keys can also be encrypted using the RSA public key, and key recovery can be accomplished using the RSA private key. The left portion of the Figure 7 indicate blockchain wallet (Metamask) platform; here, "0xe5a15B3446d830b726fEBaCF4B72614f0D9a1F62" is the sender's wallet address. Recipient platform, where receivers can download as well as upload the data file to the IPFS platform. In our implementation, the patient's data file is (DPatient) encrypted using AES (Symmetric Key) and that key (KS) is encrypted by sender's public key (PUR). The user's private key (PRR) is used to AES key recovery and encrypted file (D′Patient) converted into the original file (DPatient) which is illustrated in Figure 7.

Testing has shown that JSON, PDF, and image data file formats are among the supported file types for submitted files in our system. Figure 7 shows that a symmetric key (KS), which is encrypted using the receiver's public key (PUR), is used to encrypt the patient's data file (DPatient). The data file is retrieved from the IPFS on the receiving end using a hash value. The retrieved data file (D′Patient) is encrypted; thus, the symmetric key is decrypted using the receiver's private key, and the retrieved data file is then converted back to the original data file (DPatient) using that symmetric key (KS). The suggested system's pseudocode is shown in Figure 8. In the code provided, the term "Med" refers to a contract name that has the words "Hash" and "User" as its two data structures. IPFS hash mapping and sender-receiver addresses are included in the block information data structure. Pseudocode's "view- DataAddress" function is in responsible for returning all information associated with the sender's address, whereas "viewSenderData" produces a hash that contains the sender's address and its mapping.


0xe5a15B3446d830b726fEBaCF4B72614f0D9a1F62
DECRYPT FILE
0xe5a15B3446d830b726fEBaCF4B72614f0D9a1F62
DECRYPT FILE
0xe5a15B3446d830b726fEBaCF4B72614f0D9a1F62
DECRYPT FILE
0xe5a15B3446d830b726fEBaCF4B72614f0D9a1F62
DECRYPT FILE
0xe5a15B3446d830b726fEBaCF4B72614f0D9a1F62
DECRYPT FILE
UPLOAD
Elements
Console
Sources
top
Filter
Default levels
1 Issue: 1
Show more (1.15 kB)
Copy

```
0x23750396696139a96A662f14f81dD380b4De417c                          App.js:332
▸ NodeRSA {$options: {…}, keyPair: RSAKey, $cache: {…}}             App.js:335
private -----BEGIN RSA PRIVATE KEY-----                             App.js:291
MIIBOgIBAAJBAI8JhHJv+vyKdWZR9EOKRrB87W55S/186vNzTE0q+XLJbXJojmom
f2684ujD2ikZ0vRv6iwnl6bCAVf07At9wsECAwEAAQJAZx1PFEV0vZT3RgcwbSKQ
nUmOuxksgsv1dgDXKbclEeyzs5bMLSqifRm+YN15b/16NvxKJ4R1dClPtIA+pCf4
AQIhAJNuVT70t+pO0gfX6RHhVn0PujgvB+5lI35oA6x04WNqhAiEApsJYIKi7Uuj2
am4HypYWvJVc2YbK/ALpKCwcHQ/VFCECIQCievOL+q/F7VIaWP8906xa8LVbAekY
XZIHNNK9vtUOw0Egbo25Ty2NwtGf57/WeoA2Yp/DBpNOTvS/DLrv6HRulwECIC4b
nAeGue/t1ycADwbsi8/37YOtOwTz1D5RC0x05kf7
-----END RSA PRIVATE KEY-----
both -----BEGIN PUBLIC KEY-----                                     App.js:293
MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAI8JhHJv+vyKdWZR9EOKRrB87W55S/18
6vNzTE0q+XLJbXJojmomf2684ujD2ikZ0vRv6iwnl6bCAVf07At9wsECAwEAAQ==
-----END PUBLIC KEY----------BEGIN RSA PRIVATE KEY-----
MIIBOgIBAAJBAI8JhHJv+vyKdWZR9EOKRrB87W55S/186vNzTE0q+XLJbXJojmom
f2684ujD2ikZ0vRv6iwnl6bCAVf07At9wsECAwEAAQJAZx1PFEV0vZT3RgcwbSKQ
nUmOuxksgsv1dgDXKbclEeyzs5bMLSqifRm+YN15b/16NvxKJ4R1dClPtIA+pCf4
AQIhAJNuVT70t+pO0gfX6RHhVn0PujgvB+5lI35oA6x04WNqhAiEApsJYIKi7Uuj2
am4HypYWvJVc2YbK/ALpKCwcHQ/VFCECIQCievOL+q/F7VIaWP8906xa8LVbAekY
XZIHNNK9vtUOw0Egbo25Ty2NwtGf57/WeoA2Yp/DBpNOTvS/DLrv6HRulwECIC4b
nAeGue/t1ycADwbsi8/37YOtOwTz1D5RC0x05kf7
-----END RSA PRIVATE KEY-----
api fired ▸ {message: 'Data already exist'}                         App.js:215
Split                                                               App.js:337
  (6) ['0xe5a15B3446d830b726fEBaCF4B72614f0D9a1F62', '0xe5a15B3446d830b726
▸ fEBaCF4B72614f0D9a1F62', '0xe5a15B3446d830b726fEBaCF4B72614f0D9a1F62',
  '0xe5a15B3446d830b726fEBaCF4B72614f0D9a1F62', '0xe5a15B3446d830b726fEBaC
  F4B72614f0D9a1F62', '0xe5a15B3446d830b726fEBaCF4B72614f0D9a1F62']
Selected.............. 5                                            App.js:217
RRRRRRRRRRRRRRR ▸ [Array(4)]                                        App.js:406
response Qmat8ZnAzH2XzX0FHSaHTKvKJX8YiPPWt4Lvdf2SePCf86             App.js:409
account ▸ {message: Array(1)}                                       App.js:440
decryption started... -----BEGIN RSA PRIVATE KEY-----               App.js:441
MIIBOgIBAAJBAJ2LsaWyvk6SHZ4as0NesdNkNyMB1OTbtLRgMq+1YgIFqn3WTPIR
KozdD++1itfYLsZqBeMl6eRvYNv5B40WIccCAwEAAQJAJdrB9cAtj+1I515eAmsH
aYakFOpEwDS2R7zxkNyRTxCyBvVsnnX25huZYTfY/IRLgU7WhOHL0rDq8gBXfTy6
iQIhAMLUchAAVTaKTX+3C3xVEawT5IkxbgE2Z5c5crCmj3FtAiEAyFMSk9s+b0qcd
hS+Qf2TbP0kcoopf20V2SXYr/zfbUE4MCIQC0vp5XRqSnZ6DWRm/MTXxmJw6TYyqJ
jYtQ+xhICy/b20EgR9HQGdpYn/3Y54+pPvMhxsywzNo+ZIAkMa1KaMUNcSECIBZF
LrthSNf9SMNJq3gVBc+L6wUxP0pVvaTKe4NEERIn
-----END RSA PRIVATE KEY-----
Keys ▸ NodeRSA {$options: {…}, keyPair: RSAKey, $cache: {…}}        App.js:444
Before                                                              App.js:446
final decrypt !z%C*F-JaNdRgUjXn2r5u8x/A?                            App.js:448
D(G+KbtKw9z$C&F)J@McRfUjWnZr4u7x!A%D*GYp3s6v9y$B&E)H@McQfTjWmZq4t7w!z%gUkX
p2s5v8y/B?E(H+MbQeThVmYq3t6w9z$C&F)J@r5u8x/A?
D(G+KbPeShVkYp3)J@McRfTjWnZr4u7x!A%D*G-KaPdSgVkB&E)H@McQfThWmZq4t7w!z%C*F-
JaNdRBy/B?E(H+MbQeShVmYq3t6w9z$C&F)J@r5u8x/A?
D(G+KbPeSgVkYp3s6v9y$B&EWnZr4u7x!A%D*G-KaPdSgUkXp2s5v8y/
Bhavya                                                              App.js:452
U2FsdGVkX1/itZFI07gPRoba2DCnCtzU03xaEfUtfB2T900dGKiUaDExte6s0u7mVUjMx+3t5T
GCjEhGa3pfMtzIB54BCjAbCfjvgBJWtclwsQf8TtKr1xYUYgg6Gg j98mc9p5USzz6yMvuT+L6M
28qAC71ar35iXJ11wCDyYcD5r9JqaBQ5laNJpNvLID foRbHTvWh+ebfCr11zRhbaf95m7beHTS
```

Figure 7: Symmetric key encryption with the involvement of a public key, and data file decryption using a private key and symmetric key KS.



```
contract Med {  // contract name
    struct Hash{                // data structure
        string ipfs;
        string enckey;
        string mime;
        string description;
    }
    struct User{                  // data structure
        uint256 sent_ipfs_hash_count;
        uint256 received_ipfs_hash_count;
        mapping(address => mapping(uint256 => Hash[])) sent_ipfs_hash;
        address[] sender_address;
        mapping(address => mapping(uint256 => Hash[])) received_ipfs_hash;
    }
    mapping(address => User) public users;
    function shareData(address receipient,string memory ipfs_hash,string memory _enc_key,
                  string memory _mime, string memory _descriptions) public
 {      User storage s1 = users[msg.sender];  // hash value
        User storage r1 = users[receipient];
        Hash memory hash = Hash({
            ipfs: ipfs_hash,
            enckey: _enc_key,
            mime: _mime,
            description:_descriptions
        });
        s1.sent_ipfs_hash_count++;                r1.received_ipfs_hash_count++;
        s1.sent_ipfs_hash[receipient][s1.sent_ipfs_hash_count].push(hash);
        r1.sender_address.push(msg.sender);
        r1.received_ipfs_hash[msg.sender][r1.received_ipfs_hash_count].push(hash);    }
        function viewDataAddress() public view returns (address[] memory) {
        User storage s1 = users[msg.sender];                 // list of sender address
        return s1.sender_address;  }
    }
     function viewSenderData(address _sender, uint256 _id) public view returns (Hash[] memory)
        User storage s1 = users[msg.sender];
        return s1.received_ipfs_hash[_sender][_id];  }   }      // index
```

Figure 8: Pseudocode for proposed work in terms of smart contract.

RESULTS

Any model developer must consider the three primary security requirements of confidentiality, integrity, and availability.

Keeping information private ensures that only permitted users may access the system. Integrity ensures that communications reach their intended recipient without being altered, and availability
ensures that users can always access data when they need it.

DISCUSSION

It is a comparative analysis of the proposed work with the existing blockchain techniques in terms of confidentiality, data integrity, and access control. The proposed framework is compared against the existing blockchain-based implementations such as [11, 12, 13, 14, 15, 16]. It is apparent from Table 3 that the proposed system corrects the flaws of existing systems in terms of data confidentiality, access control, data integrity, and scalability.

Confidentiality: In this approach, the patient data file will be encrypted before being saved in IPFS. Here, the symmetric key is encrypted with the receiver's public key using the AES-CBC technique to protect the contents of the files. When a hacker intercepts, removes, or alters data being transferred between two devices, it is said to be an eavesdropping attack. Snooping, commonly referred to as eavesdropping, uses unencrypted network connections to access data being sent between machines. A brute force attack is a hacking technique that makes use of trial and error to break encryption keys, passwords, and login credentials. It is a straightforward but effective strategy for getting illegal access to user accounts. The use of eavesdropping, snooping, and brute force attacks have all been covered in our research.

Data Integrity: Data integrity means storing data that is immutable and permanent. It cannot be modified or deleted. In a blockchain, the data is stored as hash values in each block, and each block stores the hash value of the previous block in this blockchain framework. The confidence in this blockchain framework is based on the consensus mechanism, digital signature, and the designed cryptographic algorithm, in spite of relying on a third-party provider. All the blocks are associated, any alteration in the original data will result in a change in its hash value, and it is computationally difficult to tamper with the record, such that the non-tampering of the patient data is also explicitly guaranteed. The original data is stored in IPFS storage after performing a special cryptographic process.

We addressed a modification and reply attack also, which may be avoided by comprehending the encryption process. When cryptography is used, communication is typically encrypted and transmitted. The message is unlocked using the decryption method on the receiving side. The sender and receiver are required to choose a session key at random in order to prevent such attacks. A code type that will be valid for just one exchange between sender and recipient is represented by this session key. This code cannot be set up again. In addition, timestamps can be utilized. Hackers cannot crack data blocks with an expired timestamp because timestamps have a time limit.

Scalability: The proposed framework preserves most of the security requirements and provides cryptographic storage of data in IPFS, thereby resolving the scalability issue in the existing techniques. The proposed system's scalability has demonstrated and proven that it is capable of processing large datasets with low latency. Data **Security**: Data security is a vital task in our implementation as the patient data is cryptographically stored on IPFS. This blockchain framework stores only a hash of the transaction on the blockchain network, and the actual massive data is stored after encryption on the IPFS storage. This system is a patient-centric approach that provides authenticated access granted by the patient and guarantees the security of the patient's data. Also, the smart contract functionality combines with blockchain solutions to embrace high-level encryption and ensure patient confidentiality in their health care information. In addition, the data stored on IPFS is encrypted using a special AES-CBC cryptographic algorithm to establish robust blockchain data security solutions.

Table 3: Comparative analysis of proposed framework with existing blockchain techniques

Scheme	Confidentiality	Data Integrity	Data Availability	Data Security	Scalability	Blockchain Platform
Medchain [17]	Yes	Yes	Yes	Yes	No	Consortium Blockchain
Wang & Song [18]	Yes	Yes	Yes	Yes	No	Private Blockchain
Blochi [19]	No	Yes	Yes	No	No	Private Blockchain
HealthBlock [20]	Yes	Yes	Yes	Yes	No	Private-Hyperledger
SmartMedChain [21]	Yes	Yes	Yes	Yes	No	Private-Hyperledger
HealthChain [22]	Yes	Yes	Yes	Yes	Yes	Consortium Blockchain with PBFT
Proposed Work	Yes	Yes	Yes	Yes	Yes	Public-Ethereum

First, we tested our implementation on ganache, and then we used kovan, rinkeby, binance, and the matic network to simulate main net behaviour. For the Ganache platform, there is one instance that emulates the blockchain network. Here reactJS is used to build the interface for creating and viewing records, and Javascript is used to create the backend. The cryptography keys used by the patients have also been stored in Mongo DB. We have used Ethereum platform with Solidity as our test blockchain in order to simulate a blockchain network. Additionally, Web3-JS is used to communicate with the blockchain. We have used INFURA, which offers safe, dependable, and scalable access to the IPFS gateway, to test the IPFS network. In our implementation, we have tested data file upload and download times with Firebase and IPFS. We have used the file (.JSON, 147 kB) and executed it five times on a centralized database, Firebase, and a distributed database, IPFS, and measured upload and download times.

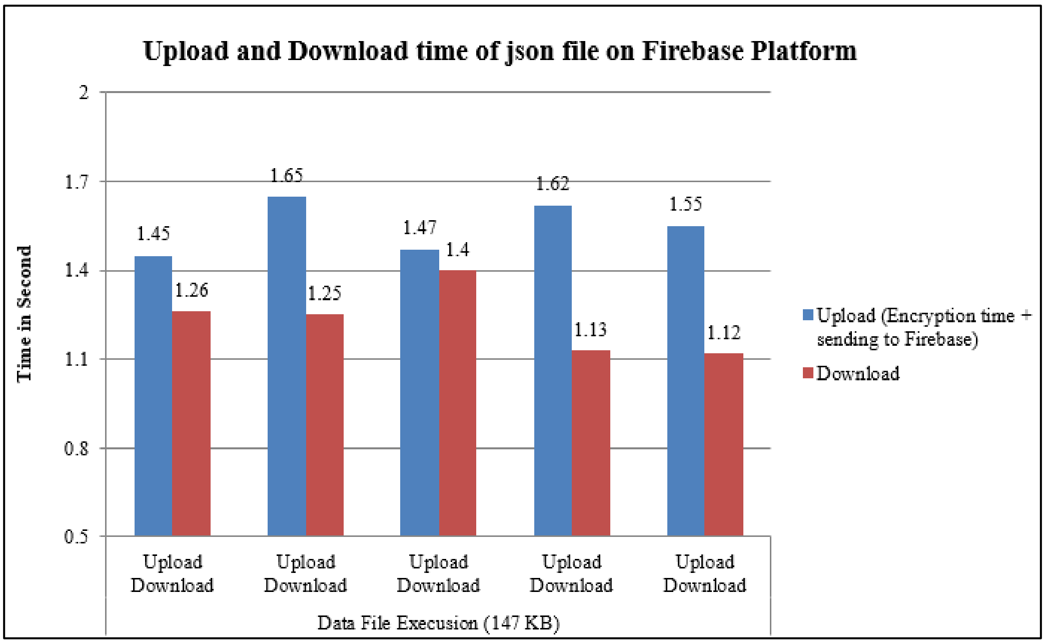


Figure 9: Upload-Download time for .json data file on Firebase.

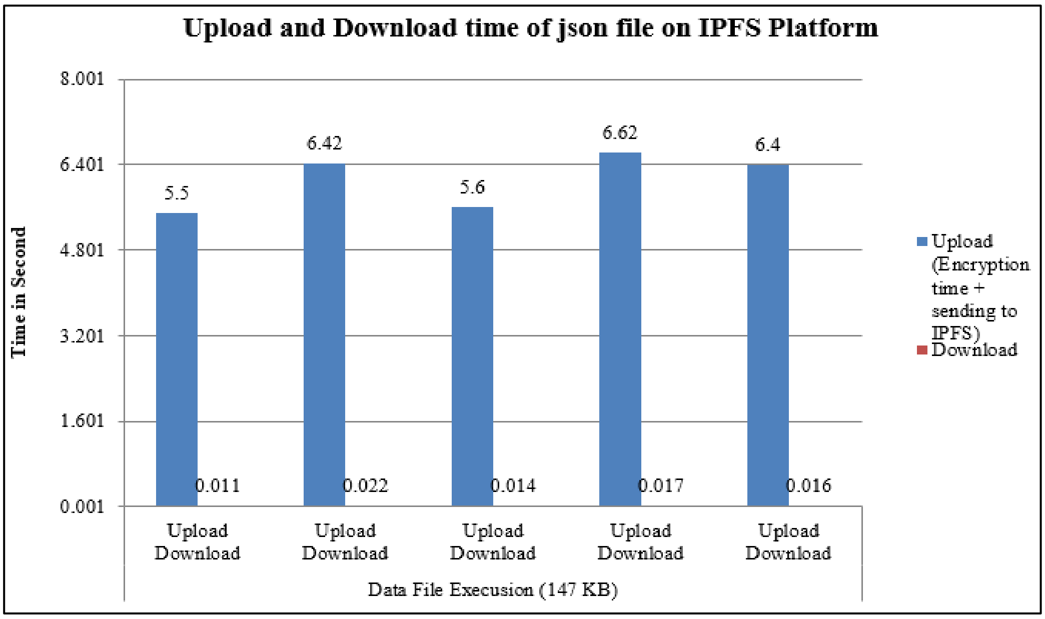


Figure 10: Upload-Download time for .json data file on IPFS.

Figure 9 indicates the results of file upload and download times for the Firebase platform, and Figure 10 indicates the time for the IPFS platform. The amount of systems storing the records, the location of the nearest system having the record, the number of systems storing the records, and other variables all affect how quickly data files may be retrieved from IPFS. This makes it impossible to accurately analyse the IPFS retrieval time for a data file.

CONCLUSION

Due to issues such as being unattended, heterogeneous, and resource-constrained, security and privacy have become some of the most significant issues in the IoT environment. In this research, we aim to demonstrate the usage of blockchain technology in the healthcare system in diverse situations, from data sharing to the development of clinical research or diagnosis aids in patient health. Patient data is generated by various IoT sensors and secured through cryptographic methods. The encrypted data could be stored on IPFS storage. The authors propose a secure model for IoT-based healthcare systems that claims to offer reliability and security.

FUTURE WORK

In our approach, we have used the IoT device Raspberry Pi to generate IoT data and applied a cryptographic algorithm for data encryption on the IoT platform instead of an innovative researcher using progressive IoT tools to support advanced encryption algorithms. The data that is already encrypted is stored in IPFS blockchain storage, which can be essentially helpful for decentralized file storage for any sort of peer-to-peer transmission. In our proposed work, we have used the Ethereum Solidity programming language for writing a smart contract. Instead of the Ethereum platform, Hyperledger Fabric, or any other platform, can be used for writing agreements between entities in the system. The proposed concepts can be applied, of course with necessary modifications, to other domains such as agriculture, smart cities, traffic management, supply chain management, etc.

ACKNOWLEDGEMENTS

We would like to thank our institute LDRP-ITR and who have helped in our research work.

REFERENCES

1. T. Alam, “mhealth communication framework using blockchain and iot technologies,” International Journal of Scientific & Technology Research, vol. 9, no. 6, 2020.
2. R. S. M. Joshitta and L. Arockiam, “A neoteric authentication scheme for iot healthcare system,” International Journal of Engineering Sciences & Research Technology, vol. 5, no. 12, pp. 296–303, 2016.
3. H. Wu, A. D. Dwivedi, and G. Srivastava, “Security and privacy of patient information in medical systems based on blockchain technology,” ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), vol. 17, no. 2s, pp. 1–17, 2021.
4. N. Dilawar, M. Rizwan, F. Ahmad, and S. Akram, “Blockchain: securing internet of medical things (iomt),” International Journal of Advanced Computer Science and Applications, vol. 10, no. 1, 2019.
5. M. A. Azzawi, R. Hassan, and K. A. A. Bakar, “A review on internet of things (iot) in healthcare,” International Journal of Applied Engineering Research, vol. 11, no. 20, pp. 10 216–10 221, 2016.
6. K. N. Griggs, O. Ossipova, C. P. Kohlios, A. N. Baccarini, E. A. Howson, and T. Hayajneh, “Healthcare blockchain system using smart contracts for secure automated remote patient monitoring,” Journal of medical systems, vol. 42, no. 7, pp. 1–7, 2018.
7. D. Rogers, “A visit to the oracle: Reviewing the state of construction industry digitalisation,” Construction Research and Innovation, vol. 10, no. 1, pp. 11–14, 2019.
8. S. Chakraborty, S. Aich, and H.-C. Kim, “A secure healthcare system design framework using blockchain technology,” in 2019 21st International Conference on Advanced Communication Technology (ICACT). IEEE, 2019, pp. 260–264.
9. T. Dey, S. Jaiswal, S. Sunderkrishnan, and N. Katre, “Healthsense: A medical use case of internet of things and blockchain,” in 2017 International conference on intelligent sustainable systems (ICISS). IEEE, 2017, pp. 486–491.
10. J. Benet, “Ipfs-content addressed, versioned, p2p file system,” arXiv preprint arXiv:1407.3561, 2014.
11. B. Shen, J. Guo, and Y. Yang, “Medchain: Efficient healthcare data sharing via blockchain,” Applied sciences, vol. 9, no. 6, p. 1207, 2019.
12. H. Wang and Y. Song, “Secure cloud-based ehr system using attribute-based cryptosystem and blockchain,” Journal of medical systems, vol. 42, no. 8, pp. 1–9, 2018.
13. S. Jiang, J. Cao, H. Wu, Y. Yang, M. Ma, and J. He, “Blochie: a blockchain-based platform for healthcare information exchange,” in 2018 ieee international conference on smart computing (smartcomp). IEEE, 2018, pp. 49–56.
14. B. Zaabar, O. Cheikhrouhou, F. Jamil, M. Ammi, and M. Abid, “Healthblock: A secure blockchainbased healthcare data management system,” Computer Networks, vol. 200, p. 108500, 2021.
15. D. El Majdoubi, H. El Bakkali, and S. Sadki, “Smartmedchain: A blockchain-based privacypreserving smart healthcare framework,” Journal of Healthcare Engineering, vol. 2021, 2021.
16. J. Xu, K. Xue, S. Li, H. Tian, J. Hong, P. Hong, and N. Yu, “Healthchain: A blockchain-based privacy preserving scheme for large-scale health data,” IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8770–8781, 2019.
17. K. M. Hossein, M. E. Esmaeili, T. Dargahi, A. Khonsari, and M. Conti, “Bchealth: A novel blockchain-based privacy-preserving architecture for iot healthcare applications,” Computer Communications, vol. 180, pp. 31–47, 2021.
18. K. Christodoulou, P. Christodoulou, Z. Zinonos, E. G. Carayannis, and S. A. Chatzichristofis, “Health information exchange with blockchain amid covid-19-like pandemics,” in 2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS). IEEE, 2020, pp. 412–417.
19. S. Singh, S. Rathore, O. Alfarraj, A. Tolba, and B. Yoon, “A framework for privacy-preservation of iot healthcare data using federated learning and blockchain technology,” Future Generation Computer Systems, vol. 129, pp. 380–388, 2022.
20. I. C. A. Pilares, S. Azam, S. Akbulut, M. Jonkman, and B. Shanmugam, “Addressing the challenges of electronic health records using blockchain and ipfs,” Sensors, vol. 22, no. 11, p. 4032, 2022.
21. P. Kumar, R. Kumar, G. Srivastava, G. P. Gupta, R. Tripathi, T. R. Gadekallu, and N. N. Xiong, “Ppsf: a privacy-preserving and secure framework using blockchain-based machine-learning for iotdriven smart cities,” IEEE Transactions on Network Science and Engineering, vol. 8, no. 3, pp. 2326–2341, 2021.
22. I. Makhdoom, I. Zhou, M. Abolhasan, J. Lipman, and W. Ni, “Privysharing: A blockchain-based framework for privacy-preserving and secure data sharing in smart cities,” Computers & Security, vol. 88, p. 101653, 2020.
23. R. Kumar, P. Kumar, R. Tripathi, G. P. Gupta, T. R. Gadekallu, and G. Srivastava, “Sp2f: A secured privacy-preserving framework for smart agricultural unmanned aerial vehicles,” Computer Networks, vol. 187, p. 107819, 2021.
24. D. Ravi, S. Ramachandran, R. Vignesh, V. R. Falmari, and M. Brindha, “Privacy preserving transparent supply chain management through hyperledger fabric,” Blockchain: Research and Applications, vol. 3, no. 2, p. 100072, 2022.
25. N. M. Kumar and P. K. Mallick, “Blockchain technology for security issues and challenges in iot,” Procedia Computer Science, vol. 132, pp. 1815–1823, 2018.
26. X. Wang, X. Zha, W. Ni, R. P. Liu, Y. J. Guo, X. Niu, and K. Zheng, “Survey on blockchain for internet of things,” Computer Communications, vol. 136, pp. 10–29, 2019.