Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
College of Information Technology

Enhancing A Security of Medical Healthcare Files Based on Interplanetary File System and Blockchain Technology

A Thesis

Submitted to the Council of the College of Information Technology for Postgraduate Studies of University of Babylon in Partial Fulfillment of the Requirements for the Degree of Master in Information Technology-Information Networks

RANA ABBAS RIDHA AL-KAABI

Supervised by

Asst. Prof. Dr. Alharith A. Abdullah

2023 A.D. **1444 A.H.**

«إِنَّ الَّذِينَ آمَنُوا وَعَمِلُوا الصَّالِحَاتِ إِنَّا لَا نُضِيعُ أَجْرَ مَنْ أَحْسَنَ عَمَلًا»

صَّدَّقَ ٱللَّهُ ٱلۡعَظِيمُ

"سورة الكهف، آية: 30"

Supervisor Certification

I certify that the thesis entitled (**enhancing a security of medical heath care files based on interplanetary file system and blockchain technology**) was prepared under my supervision Asst. Prof. Dr. Alharith A. Abdullah at the department of Network / College of Information Technology/ University of Babylon as partial fulfillment of the requirements of the degree of Master in Information Technology-Network

Signature:

Supervisor Name :Asst. Prof. Dr. Alharith A. Abdullah

Date: / /2023

The Head of the Department Certification

In view of the available recommendations, I forward the thesis entitled "**enhancing a security of medical heath care files based on interplanetary file system and blockchain technology**" for debate by the examination committee.

Signature:

Prof. Dr.Saad Talib Hasson

Head of information Networks Department

Date: / /2023

Certification of the Examination Committee

We hereby certify that we have studied the dissertation entitled (**Enhancing a security of medical heath care files based on interplanetary file system and blockchain technology**) presented by the student (**Rana Abbas Ridha**) and examined him/her in its content and what is related to it, and that, in our opinion, it is adequate with **(Excellent)** standing as a thesis for the degree of Master's in Information Technology-network.

Signature:
Name: Wael Jabbar Abed Al-nidawi
Title: **Asst. Prof. Dr**
Date: / / 2023
(**Chairman**)

Signature:
Name: Firas Sabah Salih Al-Turaihi
Title: **Asst. Prof. Dr**
Date: / / 2023
(**Member**)

Signature:
Name: Mohannad M. Al-Yasiry
Title: **Lecturer**
Date: / / 2023
(**Member**)

Signature:
Name: Alharith A. Abdullah
Title: Asst. Prof. Dr
Date: / / 2023
(**Member and Supervisor**)

Approved by the Dean of the College of Information Technology, University of Babylon.

Signature:
Name: **Dr. Hussen Atiya Lafta**
Title: Professor
Date: / / 2023
(**Dean of Collage of Information Technology**)

Dedication

Dedicated to my parents, who have continuously supported me during every step of my academic career and have always pushed me to achieve what I desire. They have been a constant source of inspiration and motivation because of their undying love and faith in me. Without their advice and assistance, I would not be in the position I am in today. To my sisters, who have supported me through all of life's ups and downs without ever giving up on me, as well as to my colleagues at work, who are always willing to lend a hand when I need it. Last but not least, I would like to express special thanks to my sister, Dr. Sura Abbas, for her constant encouragement over the previous two years as I went through the master's degree.

Acknowledgment

In the name of God, Most Gracious, Most Merciful

First and foremost, I thank God Almighty for His innumerable blessings ... My God, to you be praise and thanksgiving until praise reaches its limit ... And your praise and grace have no bounds. Praise Allah for always helping me to achieve my aims.

My great thanks and gratitude to all in the College of Information Technology at the University of Babylon, and especially to the Department of Information Networks, head, and members. Thanks and gratitude go to the honorable supervisor for his support in completing this work.

My thanks and high respect to all members of the discussion committee. in particular, to (Dr. Alharith A. Abdullah)..

I would like to express my wholehearted thanks to my family for the unlimited support, encouragement, love, and great sacrifice they provided me also for their patience, and help to me during the work.

Rana Abbas Ridha

ABSTRACT

The preservation of confidentiality for health files poses a significant challenge due to the inclusion of sensitive data. Concerns arise regarding the preservation of paper files, safeguarding privacy during the process of digitization, and the potential for unauthorized access and manipulation. This thesis presents a proposal for addressing these concerns by employing cryptographic, blockchain, and IPFS techniques, strategies, and approaches.

The proposed solution has been thoroughly tested and analyzed, confirming its validity. The approach employed leverages smart contracts and the inherent immutability of blockchain technology to enhance the security and integrity of data retrieval processes. The IPFS network enables users to securely share data. This thesis introduces methodologies that utilize blockchain and IPFS (Interplanetary File System) to achieve anonymization of sensitive data. Furthermore, provide empirical evidence showcasing the efficacy and practicality of these methods through a series of trials. The decentralized healthcare data storage system that we have implemented offers several advantages, including enhanced privacy protection. Smart contracts and the Interplanetary File System (IPFS) address the challenges associated with decentralization. The systems exhibit superior performance compared to alternative approaches in terms of storage capacity. This allows for the establishment of a private, permissioned, peer-to-peer blockchain network that includes identifiable and registered stakeholders. As a result, this network ensures optimal levels of interoperability, security, scalability, and permissioning. The Interplanetary File System (IPFS) demonstrates a high level of efficiency in the transmission of third-party data to data servers. The

system effectively addresses a wide range of privacy threats while taking into account the limitations imposed by blockchain resources. This resulted in the development of a robust and confidential healthcare data access control system, achieved through the integration of hash keys, IPFS, blockchain technology, and cryptographic algorithms. As a result, it was found that the execution time of transmitting patient file without the framework proposed would take 9.6 seconds. On the other hand, the execution time using the proposed framework, which took 2.6 seconds, Additionally, the system reduces the cost of storing files that need 0.001 ETH to 1KB.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

List of Abbreviations

Abbreviation	Description
ABE	Attribute-Based Encryption
BFT	Byzantine Fault Tolerant
CHF	Cryptographic Hash Function
CIA	Confidentiality, Integrity, Availability
CID	Content Identifier
CLI	Command-Line Interface
Dapps	Decentralized Applications
DHT	Distributed Hash Table
DHT	Distributed Hash Table
ECC	Elliptic Curve Cryptography
EHRs	Electronic Health Records
EMR	Electronic Medical Record
EVM	Ethereum Virtual Machine
GUI	Graphical User Interface
HIPAA	Health Insurance Portability and Accountability Act
IoMT	Internet of Medical Things
IoT	Internet of Things
IPFS	Interplanetary File System
IPFS	Interplanetary File System
IPNS	Interplanetary Name System
MHF	Medical Heath Files
P2P	Peer To Peer
PCIM	Patient-Centric Image Management

PHR	Personal Health Records
PoA	Proof of Authority
PoC	Proof-of-Concept
PoS	Proof of Stake
PoW	Proof of Work
SCR	Summary Care Records
SFS	Self-Certifying File System
SHA	Secure Hash Algorithm
VAT	Value-Added Tax

CHARTER 1
GENERAL INTRODUCTION

1.1 Introduction

The use of medical healthcare files has significantly improved the storage, accessibility, and legal authorization of healthcare data [1]. This has resulted in an increase in available healthcare data, which has various benefits, including disease prevention, more efficient medical care, and medical-legal culpability [2]. Furthermore, sharing and utilizing health information has led to better allocation of healthcare resources, clinical decision-making, medical quality monitoring, precision medicine, and disease risk assessment and prediction [3].

However, there are challenges related to the transparency and privacy of medical data. A decentralized storage protocol called the Interplanetary File System (IPFS) has been developed to address these issues [4]. IPFS adds a unique hash to every file, allowing users to retrieve the corresponding file even if the server is unavailable. The use of IPFS and encryption techniques ensures that patients' personal data remains secure and confidential [5]. Additionally, the use of the blockchain to track the storage and retrieval of medical data provides a record of the data's original source and retrieval procedure, offering solid proof of data authenticity [6].

Despite these solutions, providing a single platform for all healthcare participants to privately share confidential data remains a challenging problem [3]. Therefore, continued advancements in healthcare information technology are necessary to address these challenges and ensure the privacy and security of patients' personal data.

In summary, electronic medical records and health information sharing have revolutionized the healthcare industry, offering numerous

benefits for medical professionals and patients alike. However, the need for secure and transparent storage and sharing of medical data remains a critical challenge for the industry.

1.2 Related Work

There are many researchers and professionals working in the fields related to medical healthcare files, healthcare data security, and healthcare information technology. These fields are actively researched and developed by healthcare providers, technology companies, and academic institutions worldwide. Many studies and research papers are published on these topics, and numerous conferences and workshops are held to discuss and share the latest developments and advancements. Additionally, governments and regulatory bodies are increasingly involved in setting standards and regulations for the storage, sharing, and security of healthcare data. Overall, these fields are dynamic and continuously evolving, with many experts and organizations working to improve the quality, accessibility, and security of healthcare data.

In [10]Jin Sun, Xiaomin Yao et al. constructed a technique for attribute-based encryption. Their technique, which effectively restricts access to electronic medical records while maintaining retrieval effectiveness, is based on ciphertext policy attribute encryption. In the interim, they keep confidential electronic medical records in the distributed Interplanetary File System (IPFS). They used a range of operations, such as encryption, indexing, storage, and retrieval, to represent the full system.

Mohammed Moussa Madine et al.in [11] hypothesized that decentralized, immutable, transparent, traceable, and secure smart contracts built on the blockchain offer patients ownership over their medical data. The majority of existing PHR (Personal Health Records) management strategies and systems are centralized. They not just to make it extremely difficult to share medical data, but they also run the risk of creating a single point of failure. The suggested system makes utilization of reputation-based re-encryption oracles and interplanetary file systems (IPFS). Due to the anonymity all individuals and the encryption of all medical record data, the privacy of all parties, including patients, is ensured. Just one patient and their selected doctors could connect the medical records thanks to the suggested solution's stringent re-encryption mechanism, which also guarantees confidentiality .

Randhir Kumar et al. In [6] IPFS is already proposed as a blockchain-based distributed off-chain storing system for patient diagnostic reports. The problem of protecting user privacy is the underlying storage mechanism, which is immutable and content-addressable, which is a problem with the centralized model. Unauthorized access to crucial information such as identity details and diseases from which a patient is suffering, as well as misuse of patients' data and medical reports, are all threats to user (patient) privacy. Using (IPFS) and blockchain technology, the proposed system allows authorized entities, such as healthcare providers, easy access to medical data. The healthcare provider, mining process, on-chain storage, and off-chain storage are the four elements that make up the implementation. All of these modules are self-contained .

Ganesan Subramanian and Anand Sreekantan Thampy in [12] proposed a system to solve the issue of storing diabetic patients' medical records. To maintain track of medical records, the blockchain consortium is formed. Diabetes patients' medical records are secured using the Ethereum sandbox simulation concept. To preserve the privacy of personal healthcare information, the Interplanetary File System (IPFS) encrypts health data and delivers it to the blockchain. This consortium is being developed as a proof-of-concept (PoC) model using the NEM symbol blockchain. As a distributed ledger ABE method is employed to keep medical records confidential, and each stakeholder in a consortium is assigned a NEM-produced QR code to monitor records. The aims of the article are to design a framework for secure handling of diabetes patients' health data and prioritize needs during a pandemic and apply the zero-knowledge proof algorithm to validate the transaction between stakeholders such as hospitals, vaccination centers, pharmacies, government agencies, insurance companies, and other stakeholders.

Masoud Barati, et al. In [13] designed a platform for the creation of online vaccine certificates using IPFS is proposed. Only non-sensitive data is stored within the Blockchain for auditing purposes. Digital vaccine passports are one of the most important solutions for resuming travel in the post-COVID-19 world. Key challenges such as trust, scalability, and security must be overcome to implement a vaccine passport. Their proposed platform supports GDPR by implementing smart contracts. by IPFS. The DHT algorithm is implemented.

Neha Raut and Kamal Shah In [14] proposed that data tampering is one of the most serious problems in current technology. Although it may be possible to detect and forecast patients' states using data analytics

within a single entity, handling and correlating patients' related data across various organizations is difficult. The issue isn't a lack of resources; rather, it's a lack of resource management. As if to find a solution to this problem, blockchain technology is rapidly gaining attention for the security of confidential data. The main concern with this proposed methodology is protecting patients' data effectively. For this purpose, IPFS (Interplanetary File System) is used in conjunction with the Ethernet blockchain. The proposed system is developed with the help of the Ethereum blockchain, which stores patient-related data on IPFS. Now, the privacy of patients 'data is increasing. The whole control is in the patient's hand. The model is patient-centric. The patient can approve or disapprove of the doctor, as well as allow him/her to see previous histories and add new records. The use of IPFS increases its capability to store large amounts of data. In the future, the system will also arrange appointments, bookings, payments, and insurance. System integration is done using the Ethereum Blockchain.

Vinodhini Mani et al. in [15] offered an inventive approach that consists of off-chain solutions that securely store actual health data over the interplanetary file system while encrypting it in an on-chain health record database (IPFS). Due to privacy, confidentiality, and security concerns, the development of blockchain-based electronic health systems is constrained. They do this by describing PCHDM, an end-to-end secure health record chain network architecture, and its design, implementation, and evaluation. To guarantee the security of health records amongst stakeholders, the architecture combines networks, IPFS, and smart contracts The system that has been put into place seems effective and meets several security standards. It is possible to achieve a high level of confidentiality, transparency, and reliability.

Driss El Majdoubi et al.in [16] the researcher proposed , the Internet of Things (IoT) is changing the healthcare industry through accelerating sharing, including the use of client records and incorporating service users. Though they can access and exchange their private health data from anywhere, patients are now more satisfied and motivated with IoT-enabled devices. That new approach makes medical care provision increasingly feasible by enabling machine-to-machine connectivity, interoperability, data mobility, and medical interchange. To preserve privacy in data sharing in an s-healthcare system, they created and deployed Smart Med Chain, an end-to-end Blockchain-based architecture. Patients' medical IoT data will be transmitted and monitored, and clinicians can access it with their permission. To guarantee scalability, they only retain the hash of health records on the blockchain; the actual data is saved after encryption in the distributed storage system IPFS. The analysis' conclusions show that the suggested solution is workable and satisfies many regulatory standards. The likelihood that it will maintain health data security, transparency, confidentiality, consistency, and flexibility is the highest

Muhammad Mohsan Sheeraz et al. in [17] presented a blockchain-based architecture for gathering healthcare data. In the first phase, the participants for data collection will be registered on the blockchain. In the second phase, they will be collected using a software application. Then the data will be encrypted and stored on IPFS after the identity verification of the data sender. After successful storage of data, the data index of IPFS will be stored on the blockchain network. In this way, only authorized participants can participate in data collection, and accurate data will be collected. The data stored on IPFS is secured and can only be identified

by the indexes that are stored on the blockchain. Since healthcare data is sensitive, they used encryption techniques to encrypt the data. Every user in the system has to register on the network, and a private and public key pair will be issued to the users. These key pairs will be used as credentials to interact with the system. They have used the dApp concept to make data consistent and structured.

In [8] Kebira Azbeg et al. presented a chronic disease management system based on IoT, Blockchain, and IPFS technologies. This method has a number of advantages in terms of remote patient monitoring. It collects, shares, and protects data on a daily basis. There are three pieces to the system. The first side is in charge of data gathering. To ensure collection, this side is utilizing IoT healthcare gadgets. The second side is in charge of safely sharing data. Blockchain is the technology that makes this possible. The final side is for data storage, and it employs IPFS. Any healthcare system can benefit from thesystem. However, in the situation, they chose to employ it, particularly in the treatment of chronic disease systems, because this type of condition requires daily follow-up and regular check-ups. The suggested system is completely decentralized, and it provides a high level of security by utilizing Blockchain, smart contracts, proxy re-encryption, and IPFS to regulate access to patient data, preserve privacy, and assure data integrity (Clique PoA algorithms, PoW algorithm).

Battah et al. in [18] proposed a system consisting of entities that communicate with the smart contracts to govern the access control of the encrypted data stored on the IPFS. By securing the information with a cryptographic algorithm and delivering it to the P2P decentralized repository in with another key encrypted by the public key of a shared

wallet among authorized users and the proprietor of the data utilizing multi-signature, the suggested scheme maintains confidentiality. In addition, the creator of the data generates a smart contract that includes the hash of the aforementioned parts, which serves as the data's address. The owner of the data then generates a re-encryption key using its private key and the public key of the requestor to submit to the proxy servers. The data is downloaded by the client application from the proxies. It then goes on to decrypt both the data and the symmetric key has used its private key before decrypting the data once and using more than the symmetric key .

Mohamed Yaseen Jabarulla and Heung-No Lee [19] In the field of health care, medical images are stored and exchanged. Current procedures rely on cloud-based centralized space and cause privacy problems when sharing data over a network. The researchers presented a proof-of-concept architecture for the proposed PCIM (patient-centric image management) system, which is a decentralized framework based on the Ethereum blockchain and IPFS for storing and distributing medical pictures. It is designed for a distributed patient-centered image management (PCIM) system that aims to keep data safe and control it without relying on a central system. Use the PCAC-SC management system, which allows authorized entities to access blockchain data. They encrypt the sensitive medical images before uploading them to the global IPFS network. This ensures data originality, ensures data security, and prevents data from being leaked to irrelevant users. A pair of asymmetric keys, a public and a private one, is generated.

In [20] Mohamed Yaseen Jabarulla et al. proposed a decentralized solution for storing and sharing medical photos based on blockchain and IPFS. Before being posted to IPFS, the image files are encrypted using

steganography and asymmetric encryption. An open asymmetric encryption approach hashes and protects the image content. In addition, we encode the patient's description on medical photographs using steganography technology.

Raghavendra K. Marangappanavar and Kiran M.[21] Data security issues have made it difficult to share health information, which could jeopardize patient privacy. Health record management and security techniques now in use have been shown to be insufficient. They proposed an architecture for a decentralized blockchain-based PHR sharing mechanism that ensures anonymity, taking advantage of emerging technologies like IPFS. The idea shows how to use a smart contract and an access control system to adequately preserve data that may be shared with patient authorization. The system effectively functions as a multiple-access system because healthcare providers have had their own records. The system protects data and information for data protection and adherence to fundamental health sector requirements. To deploy smart contracts, a truffle suite is utilized, which provides contract addresses for contract calls. Depending on the type of request, the data owner assigns a process for translation. A transaction is created and authorized using just a private key.

Randhir Kumar and Rakesh Tripathi. In [22] To address concerns with the privacy risk of COVID-19 patients' information, including unauthorized access to sensitive patient data like specific results and clinical records, they proposed a distributed on-chain and off-chain storage model based on consortium blockchain and interplanetary file systems (IPFS). Large amounts of COVID-19 patient records can be

potentially saved because of peer-to-peer file storage models made possible by the Interplanetary File Systems (IPFS). The underlying storage mechanism retains a content-addressed hash of the files and uses distributed hash table (DHT) and version-control methods to get rid of duplicate files.

Abdullah Al Mamun et al. In [23] suggested a framework for EMR in the healthcare sector that combines a blockchain and the Interplanetary File System. Electronic medical record (EMR) systems confront significant concerns with data management, security, and accessibility. Unauthorized access to medical records and improper use of patient disease reports are among the many security dangers to patient privacy. Prior to submitting files to the blockchain network, the suggested solution additionally intends to minimize record volumes. Additionally, a distinctive IPFS hash and patient control over the EMR provide data immutability. The client must obtain the private key from the data owner in order to view the patient's medical records. The AES-256 method is used to encrypt the EMR. Additionally, it offers total control over the data.

Randhir Kumar and Rakesh Tripathi In [9] proposed that the Internet of Medical Things (IoMT) is the next frontier in the digital revolution, and it leverages IoT in the healthcare domain. However, according to cloud-based storage, IoMT poses a significant issue for data storage management, reliability, and transparency. They suggest a consortium blockchain network with smart contract support to solve these problems. On order to initially implement smart contracts for patient and medical device authentication in the same cluster layer, they integrate (IPFS) cluster nodes. The primary goal of this research is to offer a layered

architecture for the authentication and storage of medical devices that can prevent different security and privacy issues in IoMT-enabled healthcare. The suggested model is split into two sections: In order to protect the privacy of patient data, registration, medical device authentication and authorization, and information distribution in the blockchain network. The suggested paradigm resolves current issues and improves how the IoMT healthcare network operates. Distributed off-chain storage, which is highly secure and protects privacy, is the foundation upon which the paradigm is created and implemented. The suggested approach makes IoMT healthcare systems more scalable and enables secure access to patient data by utilizing an IPFS cluster. The Solidity programming language (version 0.4.26) and the Remix IDE were used in the research. The IPFS cluster node which it also immediately communicates with the application interface and guarantees device verification and the storing of their addresses is where the smart contracts are installed. The all related works papers summarize in Table 1.1.

Table 1.1 Blockchain and IPFS-Based MHR Data Security Methods

Papers	Year	Techniques used	Advantages	Disadvantages	Implementation
[10]	2020	Ethereum blockchain, IPFS	○ MHR data retain a reliable, safe, and unchangeable audit trail that anybody can check. ○ Fully decentralized.	○ The audit trail of MHR data is reliable, unchangeable, and secure so that anyone may check it. ○ are really difficult. ○ There is no key exchange mechanism.	Remix Solidity IDE
[11]	2020	Blockchain	○ Secure content storage ○ Verifiable keyword ○ Search access control	○ access privileges and the timeliness of expired users ○ functional problems with blockchain data.	N\A
[6]	2020	Ethereum blockchain, IPFS	○ give patients control over their medical records in a decentralized, traceable, reliable, trustful, and secure manner.	○ patients do not have full control over their data because it is stored in hospitals. ○ Interoperability ○ Key management ○ GDPR ○ Smart contracts upgradability	Remix Solidity IDE

[12]	2020	consortium blockchain and IPFS based off-chain storage model	○ provide privacy of the patient reports	○ transaction upload is more computation-intensive than ○ transaction download for all report sizes.	Python
[13]	2021	The NEM symbol Blockchain, IPFS	○ secure handling of diabetes patients' health data ○ keep medical records confidential	○ costly drugs ○ knowledge of diabetes disease ○ COVID-19 and diabetes. ○ existing healthcare system for diabetes ○ patients during covid_19	N\A
[14]	2021	blockchain, IPFS	○ keeping and verifying patient data	○ sharing vaccine passport data between different organizations, regions, and countries	Solidity language.
[15]	2021	blockchain, IPFS	○ protecting personal data and enabling citizens to control creating, storing, and verifying digital vaccines certification	○ implementation of both access control ○ and encryption management layers of the designed architecture. ○ development of the proposed platform ○ in the cloud environment and the management of CIDs	Solidity language. Ganache is a local test network
[16]	2021	Web3.js, Ethereum blockchain, IPFS	○ protect patient's data from different illegal access. ○ Patients have confidentiality towards their record	○ Data tampering ○ correlating patient-related data due there is lack of resource management	Ethereum Blockchain (Ganache), Solidity

[17]	2021	Hyperledger Fabric Blockchain, IPFS	○ Privacy, security, integrity, interoperability, and scalability are issues with patient-centric distributed architecture when storing patient-centric data. ○ They have developed a revolutionary algorithm for utilising blockchains to securely store and access records. ○ To ensure scalability and effectiveness, the initial massive amounts of data are maintained off-chain in IPFS.	○ The implementation of multi-blockchain systems calls for an enormous amount of resources. ○ The system included Non-Fungible Tokens (NFT) so that stakeholders could access audio and video as NFT data.	Node.js, Java, access control languages
[8]		SmartMedChain,blockchain ,IPFS	○ Has the capability to assure security, privacy, confidentiality, integrity, and scalability of the health data and is effective in practice while meeting various security standards.	○ In a broad smart healthcare ecosystem, using many Blockchains may need a lot of resources.	Node.js web service API

[18]	2022	Blockchain dApp, IPFS	○ keep data accurate, consistent, reliable, and easily accessible to the researchers. ○ ensure authorized access to the data. ○ IPFS provides secure sharing and easier. ○ accessibility of data.	○ The process of collecting data for research is always a difficult and time taking process. ○ the data is unaccusable. ○ distributed, unstructured, inconsistent, and complex.	theoretical research
[19]	2022	private Ethereum Blockchain, Clique PoA, IPFS, proxy re-encryption	○ daily data collection, data sharing, and security ○ system is fully. ○ decentralized, and it offers a high security level	○ Computational, ○ Mobility ○ Access control and data leakage.	Remix IDE.
[20]	2020	novel proof-of-concept design with blockchain and IPFS	○ allows users to have full ○ control of their medical images by ensuring guaranteed security, transparency, and data ○ integrity ○ The use of IPFS in medical image migration time and retrieval time is faster	○ due to the decentralized nature of their systems, such as losing private keys.	Solidity is a programming language that is integrated into the Remix IDE.

[21]	2019	The ciphertext policy attribute-based encryption system and IPFS storage environment, combined with blockchain technology	○ Enabling the provision of secure sharing of medical images across domain networks. ○ store user information on the blockchain ledger. ○ the authentication layer performs decryption and ○ verifies the authenticity of the image.	○ technologies for transferring medical images are inadequate owing to maintenance cost, privacy, storage, and security concerns.	N\A. .0
[22]	2020	Web App, Blockchain, IPFS	○ Putting data in an IPFS hash for quicker retrieval that maintains duplicates everywhere to prevent a single point of failure.	○ Security and privacy of medical data ○ Performance ○ Scalability Energy ○ Consumption	Implement a smart contract on a decentralized blockchain platform using Solidity.
[23]	2020	consortium blockchain network, IPFS	○ provides immutability and keeps privacy of the patient's records	○ the model need more numbers of peers ○ and multiple sizes of megabytes of reports sharing system.	solidity
[24]	2021	blockchain, IPFS	○ protecting patient privacy, allows convenient access by approved authorities such as healthcare providers to medical data	○ misusing patient disease reports, ○ unlawful access to medical records.	Python

[9]	2021	smart contracts enabled consortium blockchain network, (IPFS) cluster node	○ The decentralized nature of the system is guaranteed by the blockchain-based architecture. ○ For the patient and their medical devices, the registration-based security paradigm is described. ○ To maintain anonymity in the IoMT network, the access control is created and executed utilizing consortium blockchain.	○ Utilizing IPFS to create a distributed cluster. ○ So much to process complexity is needed to maintain more devices in the system.	node js, solidity version 0.4.26 and remix IDE

1.3 Problem Statements

The management and storage of patient information, whether in paper-based or electronic health files (EHFs), present significant challenges. In the traditional paper-based system, there are risks of physical damage, loss, and unauthorized access. The transition to EHFs introduces new concerns, including privacy protection, secure storage, prevention of malicious attacks, and unauthorized access. Patient privacy is at risk during the conversion from paper to electronic records, and unauthorized access can lead to identity theft and fraud. Securing EHFs is crucial to prevent breaches and manipulation of records, such as through hacking or ransomware attacks. Additionally, the lack of interoperability and data sharing mechanisms can hinder the efficient retrieval of medical records.

1.4 Thesis Aims

The aim is to enhance a secure and efficient systems for managing electronic health files that can ensure the protection of patient privacy, prevent unauthorized access, and enable secure sharing of health information between authorized parties. The use of technologies such as blockchain and the (IPFS) have been proposed as potential solutions to these problems, as they offer the benefits of decentralized storage, immutability, transparency, and traceability. By implementing these technologies in healthcare systems, it may be possible to improve the efficiency and security of health file management, while also enhancing patient control over their own health data.

1.5 Thesis Objectives

- Improve data integrity: By utilizing blockchain's distributed ledger technology, the integrity of health files can be ensured by maintaining a tamper-proof record of all changes made to the record. IPFS can be used to store the actual files in a decentralized manner, making it harder for malicious actors to manipulate or corrupt the data.
- Enhance security: Blockchain technology can provide a secure framework for storing and sharing health data, using encryption and access controls to limit access to authorized users only. IPFS can ensure the data is stored securely and remains available even if some nodes go offline.
- Ensure privacy: The sensitive nature of health data means that privacy is a critical concern. The use of blockchain technology and IPFS can help to ensure that patient data is only accessible by authorized parties, and that any changes made to the data are transparent and auditable.
- Increase accessibility: Electronic health records can be made more accessible by using blockchain and IPFS, allowing patients and healthcare providers to access records from anywhere in the world.
- Reduce costs: By using a decentralized and secure system for storing and sharing health data, costs associated with maintaining and securing traditional health records can be reduced, potentially leading to more efficient and cost-effective healthcare services.

1.6 Thesis contribution

The presented system offers an efficient and secure approach to managing medical health files (MHF) through blockchain technology. By integrating blockchain with IPFS, existing MHF systems can be enhanced, addressing multiple storage issues. The system prioritizes privacy and security measures, utilizing a re-encryption scheme to protect the confidentiality of patient files and ensure they are only accessible by authorized users. Additionally, the system enables safe retrieval of files using IPFS technology, further enhancing the overall security and efficiency of MHF management.

1.7 Thesis Organization

The remaining chapters of this thesis are organized as following:

Chapter Two: It presents the theoretical background, including the concepts of medical health care, blockchain technology, IPFS protocol, and security files , Chapter Three: This chapter introduces the proposed system and its implementation. , Chapter Four: The fourth chapter presents and discusses the evaluation and obtained results. And, Chapter Five: The last chapter of this thesis states the conclusions and suggests several future works.

CHAPTER 2

THEORETICAL BACKGROUND

2.1 Overview

This chapter describes the contents of the thesis. The main components include medical files and the techniques used in the proposed system, such as Blockchain and IPFS, in addition to an explanation of the tools and ways to use them for connection and achieving goals. The present chapter illustrates an overview of this technology and describes some basic terminologies that have been used. It provides a description of the background, which is related to the current work with an emphasis on Blockchain.

2.2 Medical Health Files (MHF)

Medical health files refer to the collection of records, documents, and information related to an individual's health history and current health status. These files may include details such as medical history, current medications, test results, and doctor's notes, among others. These files are important for healthcare providers to have a complete understanding of a patient's health and to provide effective treatment. In some countries, patients have the right to access and control their medical health records[24].

2.2.1.Medical health files classification

Medical health files can be classified into several types, including:

- Electronic Health Records (EHRs): Digital version of a patient's health information, including demographics, medical history, test results, and medications[25].
- Physical Health Records: Paper-based health records, typically kept by a healthcare provider or hospital.[26]

- Personal Health Records (PHRs): Health records that individuals can access and manage themselves, either through a website or app[27].
- Summary Care Records: A brief overview of a patient's health information, including current medications, allergies, and important health conditions, designed for use in emergency situations[28].
- Imaging Records: Digitized copies of medical imaging tests, such as X-rays, CT scans, and MRI scans.[29]
- Genetic Health Records: Records of genetic test results and family health history.[30]

These are some of the common types of medical health files. The specific type of file depends on the individual's health status and needs, as well as the country's healthcare system. As Shawn in (figure 2.1)

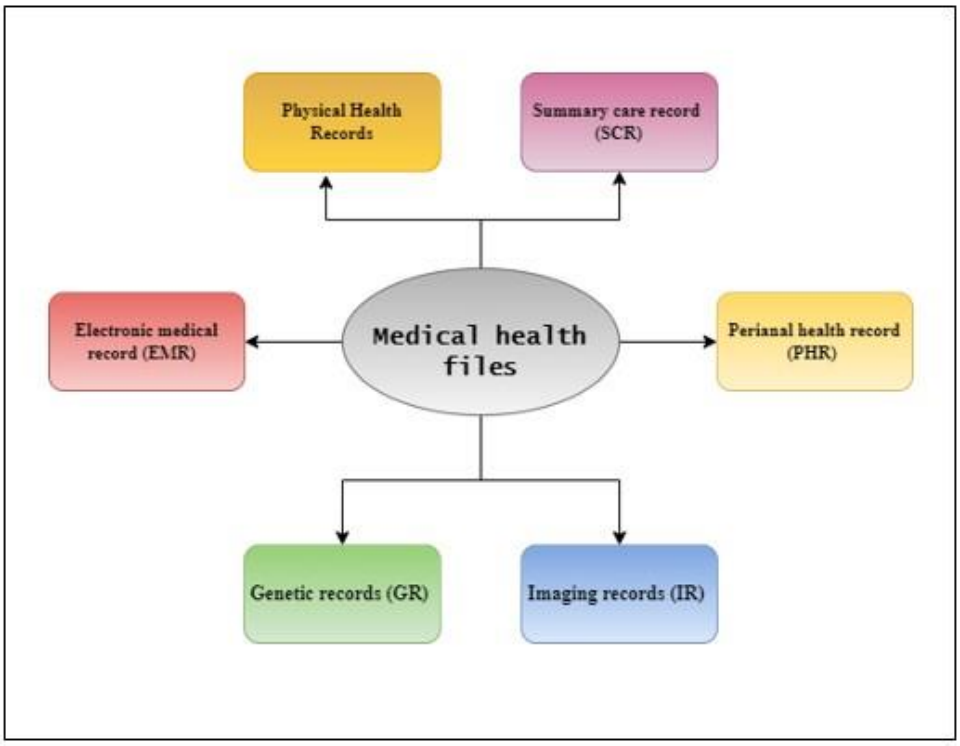


Figure 2.1 Medical health files classification.

2.3 Healthcare security Application

The United Nations described health security for the first time in 1994. Many references after that have used the term "health security" to describe health problems that directly impact human security. Public health security, global health security, international health security, and global public health security are all commonly used [32]. Although information security is a top priority for all organizations, healthcare providers must be vigilant in protecting sensitive patient information. Government regulations, such as the US Health Insurance Portability and Accountability Act (HIPAA), create privacy protections for protected health information, in addition to the emerging threat posed by hackers and other intruders. The development of a network firewall alone is inadequate. Instead, providers must take a holistic approach to safeguard patient data at all points of entry, both within and outside the network.

Healthcare organizations are becoming more vulnerable to nontraditional attacks as they rely on networks for their core operations [33].

2.4 Blockchain Technology

The Blockchain is a distributed database that stores all data securely, transparently and verifiable. Blockchain is a digital transaction arranged in chunks of data called blocks; it is linked by chain via a cryptographic validation called hashing function that forms an unbroken chain. A Cryptographic Hash Function (CHF) of the previous block is included in each new block [34]. Blockchain is programmed not only to record financial transactions but also for everything that has value. Blockchain is also called Distributed Ledger [35]. The most popular Blockchain implementation is the Bitcoin (BTC) to handle crypto-currency that proposed in 2008 by Satoshi Nakamoto [36]. Blockchain operates in a P2P network, meaning each node in the network has a full copy of the information. No single node can control the network, thus removing the central authority over this database[37]. The peer to peer P2P architecture of Blockchain technology increases the toleration of error. Even if some peers are removed from the network, they still work in a normal way. Furthermore, since blocks cannot be changed without changing the complete chain of blocks, it makes the system more flexible and increases the difficulty for the attacker [38] Modern researches exhibit that Blockchain technology is an efficient solution for issues such as unsecured data storage, high cost, and low efficiency. Bitcoin, Ethereum, and Hyperledger Fabric are just a few of the most well-known and representative blockchain platforms [39].

2.4.1 Component of Blockchain Structure

The Blockchain system generally consists of a number of peers, each owning a local duplicate of a distributed ledger. These nodes do not need a central authority to confirm and coordinate transactions. However, they communicate with each other to obtain an agreement on the content of the ledger [40]. A block header and the body header make up the block .

The most important components are explained as follows:

1. **Transaction:** In Blockchain, the transaction represents the procedure the user launched on the network. It could be recording information that a Blockchain-based system deals with [41].
2. **Block:** is the set of valid transactions and other details. In the Blockchain, any peer can initiate a transaction and broadcast it to all peers in the network. Network peers validate the transaction using the old transactions, the moment that the transaction is validated next step is added to the existing Blockchain. It can be divided into two parts, block header and block data [42]It is worth noting that each Blockchain can define its block fields; Many Blockchain s contain the following fields: [43].
   A. Block Header: The Blockchain comprises blocks, each of which has a complete record of all the transactions that have ever occurred in that particular block. A block has just one parent if its header includes a hash of the block before it. The first block in a Blockchain is called the genesis block, and it does not have any parents. These fields of the Block header are summarized as follows:
   - The block number: It represents the block's sequence in the Blockchain.

- The previous hash block: The Blockchain system uses the previous hash to create the new block's hash, making the Blockchain tamperproof.
- The current hash block can be accomplished in various ways using the hash of the fully integrated block information.
- A timestamp is a recorded unit of information that records the order in which transactions occur in blocks, given by time reference.
- The block size: which determined by protocol rules applied in each Blockchain.

B. Block Data: These fields of the Block Data are summarized as follows:

- A list of transactions.
- The number of transactions.
- The number of validated transactions in each block.
- After validation, the block is distributed to all participants in the network. The first block in any Blockchain is called a Genesis Block[42].

3. **Mining:** is the process of appending a new block (transactions) to the Blockchain. The Blockchain relies on the miners (specific nodes in a network) to aggregate valid transactions into blocks and append them to the Blockchain. New blocks are broadcast across the entire network, so each node contains an exact copy of the entire data structure [44].
4. **The Consensus Algorithm:** is used in the Blockchain to solve the problem of guaranteeing data consistency in various failure peers in a distributed system. Consensus mechanisms allow distributed systems to work together and stay secure. There are several consensus mechanisms used in different Blockchain networks. The most famous

of them is Proof of Work (PoW) is adopted in Bitcoin, and Proof of Stake (PoS) is adopted in Ethereum. The main advantage of PoS over a PoW is that PoS uses much less electricity to run and is thus more cost-effective[45].

2.4.2 Layers of Blockchain

Blockchain architecture generally consists of six-layer:

- Data Layer: This layer specifies the essential structure of data, including digital activities, blocks, and cryptographic keys, arranges them into Blockchain s, transaction pools, and wallets, and manages a wide range of data functions (read/write/cache/encrypt/decrypt) [46].
- Network layer: The technology of point-to-point transmission (P2P network technology is another name for peer-to-peer network technology), propagation mechanisms, and verification methods are the major components of this technology. Consensus techniques, encrypted signatures, data storage, and other features are included. The network layer's main goal is to create a chain of information communication between nodes in a network [47].
- Incentive layer: The main purpose of the incentive layer, which combines economics with Blockchain technology, is to offer incentives to encourage other blocks to check the security of the Blockchain and to get people to help with the computing power[48].
- Smart Contract Layer: The contract layer encompasses a variety of script codes, algorithmic processes, and smart contracts that create regulated and auditable contract specifications. Smart contract flaws include a disordered exception, reentrancy, dependency on

timestamps, reliance on block numbers, appeal for a damaging delegate, and freezing, to name a few. Hackers can easily exploit smart contracts owing to faults and weaknesses. A single trusted verifier or a group of trusted verifiers can confirm a smart contract. Smart contract development, on the other hand, lacks discipline and consistency. Program testing can be performed to discover the presence of bugs. However, it is unable to determine whether or not bugs exist. Given the financial nature of smart contracts, vulnerabilities or faults in their systems could have disastrous effects. On the other hand, smart contracts can benefit from the formalized process. For example, it may be able to detect a variety of errors and inaccuracies in existing semantics. It may also be used to mathematically determine whether the code fulfils the specifications.[49].

2.4.3 Blockchain Consensus Layer Algorithms

In this section, a brief explanation of the most prominent blockchain algorithms is provided.

- **Proof of Work (POW):**The combination of encryption and processing power in a Proof of Work (PoW) algorithm establishes consensus and ensures the authenticity of data stored on the Blockchain. After proving the validity of a block, network nodes (known as miners) use their computational power to validate transactions (ensure that a sender has sufficient funds and is not engaging in double-spending) and, more significantly, compete in a race to solve the protocol's cryptographic challenges[50].

- **Proof of Stake(PoS)** One of the most promising strategies for replacing PoW while maintaining similar resilience qualities is Proof of Stake (PoS). Despite the fact that PoW necessitates the honesty of a (qualified) majority of computer power, PoS assumes that honest participants control the majority of the money in the system. Individuals with significant interests in the system have a financial motive to keep it working according to the protocol specifications because they risk losing their shares if the coin loses trust [51].To accomplish the leader's election and maintain network consensus, PoS uses virtual resources such as a node's stake. Because the mining resources are virtual, the PoS-based consensus process is instantaneous and has no costs. However, some attacks that specifically target POS protocols using voting mechanisms are far-reaching[52].
- **Proof of Authority (PoA):** is a permissioned blockchain consensus algorithm family that has gained traction due to its superior efficiency over classic Byzantine Fault Tolerant (BFT) algorithms, especially when handling message changes. PoA is initially proposed within the Ethereum ecosystem [81] for use within closed networks. Since the standard Ethereum protocol is based on PoW, it can fork if two competing chains decide to append blocks to the same index. This forking condition can cause security problems like double spending if it is not discovered quickly enough.
- PoA protocols, an alternative protocol designed to prevent forks, were recently integrated into the most widely deployed versions of Ethereum, parity and Geth, and are now used all over the world. PoA's popularity has increased, and it is now utilized by a number of blockchain networks and offered by numerous sizable SaaS vendors[53].

2.4.4 Blockchain Features

Blockchain technology has several key features, including:

- Decentralized Data Management: Each user owns the authority to add data to the Blockchain; thus, no single user owns the system more than any other user[54].
- Immutability: Blockchain is a way to store information that cannot be changed or tampered with. A unique cryptographic hash is used to check the data in the Blockchain. The previous block's hash links the new block to the one before it. Even if changes are made to the block, the next block will still have the previous block's hash. So, the hacker must change all blocks to hide the change to one block [55].
- Transparency: Every information or transaction on the Blockchain is public, which allows each node in the Blockchain network to access all information or transactions without tampering with it; this makes the system transparent.[56].
- Disintermediation: When middlemen like banks are taken out of transactions, costs and risks related to the existence of this middleman go down.[25].
- No Risk of Central Failure: Usually, the central server stores the big data. However, users do not control their data. The decentralized storage of the Blockchain is not stored in one location, and that means keeping a copy of the data on every peer in the network.[57].

Redundancy: All Blockchain nodes keep a copy of the information on a P2P network. This means that a malicious action cannot change it (attacker). Therefore, if a hacker wants to change any information in a

Blockchain, he or she has to make the same changes to all nodes in the network. This takes a lot of computing power[57].

2.4.5 Blockchain Classification

Blockchain technology comes in many forms, but the most important ones are [58].

- Public Blockchain: They are chains that anyone can join and have a say in what happens. In this type, no participant has a ledger because it is accessible to everyone. Instead, the instructors use a decision-making method called "distributed consensus" to keep a copy of the ledger on their contract.[59].
- Private Blockchain: This kind is not available to everyone. Only a certain group of people can access it, and only those can see the ledger.[60].
- Consortium blockchain: A consortium blockchain is a decentralized network that is controlled by a group of organizations. Access to the network is restricted to authorized participants, but unlike private blockchains, transactions are usually public[58].as shown in (figure 2.2)

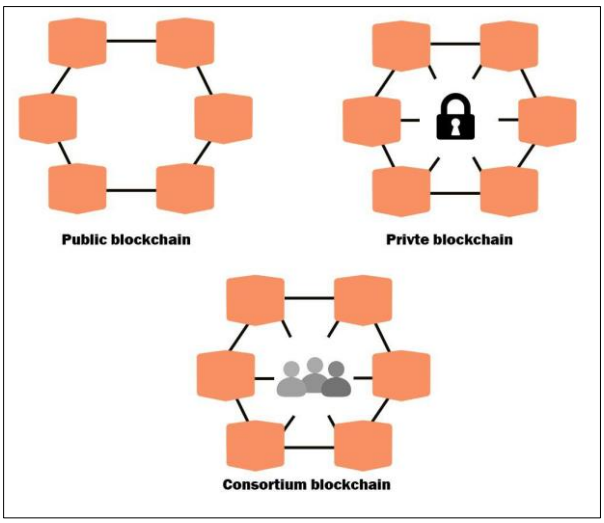


Figure 2.2 Classification of blockchain.

2.4.6 Health care Application in Blockchain

Blockchain technology for health record management systems could play an essential role in healthcare management to achieve more transparency over patients' health data and to share them between hospitals, making it easier to know the patient's health history [66]. Anti-counterfeiting detection of drugs to see if the drug is counterfeit or original using Blockchain technology, a medical product can be verified as fake by tracking them from the origin Identity verification: Blockchain can be used to create secure and decentralized systems for identity verification, which can be used for everything from voting systems to online marketplaces[65].

2.4.7 Smart Contract in Blockchain

A smart contract is a part of executable code that runs on the blockchain that helps to facilitate, carry out, and enforce an agreement's conditions. A smart contract has an account balance, private storage, and

executable code. The contract's state comprises the storage and the balance of the contract[68]. The state is stored on the blockchain. and it is updated each time the contract is invoked. Once the contract is deployed on the blockchain, the contract code cannot be changed. To run a contract, users can simply send a transaction to the contract's address[69]. This transaction will then be executed by every consensus node (called miners) in the network to reach a consensus on its output. The contract's state will then be updated accordingly. The contract can, based on the transaction it receives, read/write to its private storage, store money into its account balance, send/receive messages or money from users/other contracts, or even create new contracts[70]. When the predetermined criteria are satisfied, a smart contract's primary goal is to automatically carry out the terms of an agreement. In contrast to conventional systems that demand a reliable third party to enforce and carry out an agreement's terms, smart contracts promise to have lower transaction costs[71].

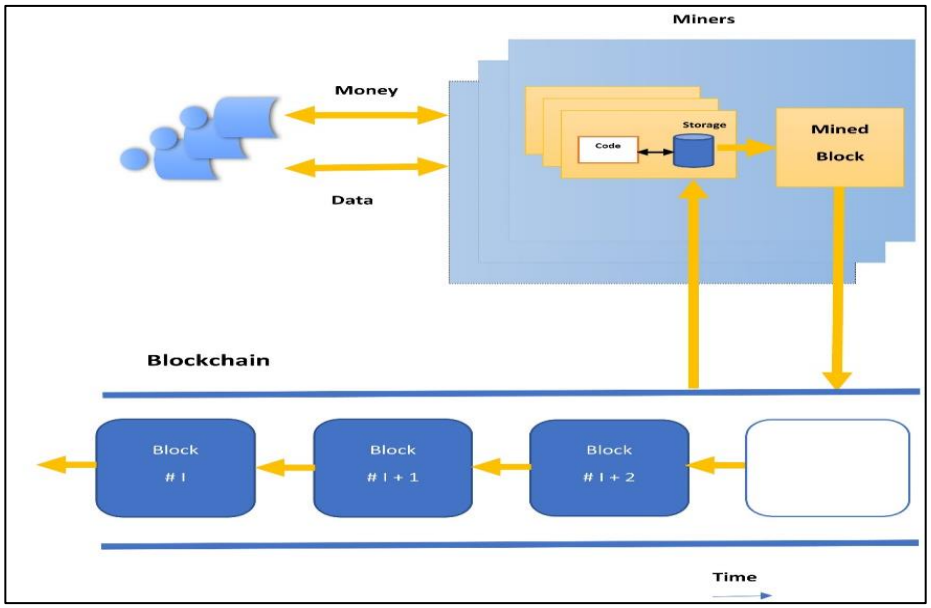


Figure 2.3 Stricture of smart contract in blockchain.

2.5 Interplanetary File System (IPFS)

IPFS (Interplanetary File System) is a peer-to-peer protocol and network designed to create a content-addressable, distributed method of storing and sharing hypermedia in a distributed file system. IPFS is initially designed by Juan Benet, and is now an open-source project. It aims to make the web faster, safer, and more open by replacing the traditional, centralized model of the web with a decentralized one. This allows for greater resilience and censorship resistance, as well as the ability to share large files more efficiently[72]. (As shawen in figure 2.4)

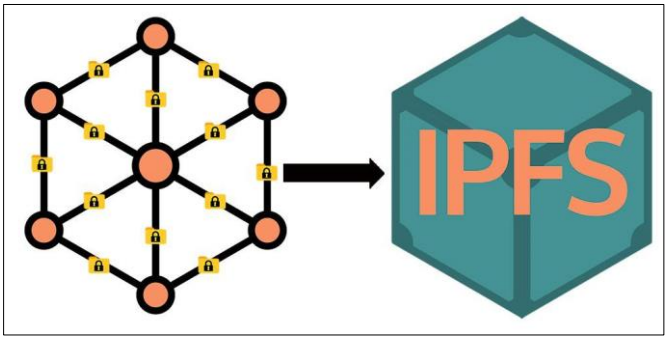


Figure 2.4 IPFS[72].

2.5.1 The Concept of IPFS

IPFS works by breaking files into smaller pieces called blocks, and then creating a unique, content-addressable identifier (or "hash") for each block. This allows for efficient distribution of the file, as well as the ability to verify the integrity of the data[7].When a file is added to IPFS, it is broken into blocks and each block is given a hash. The hashes of the blocks are then used to create a Merkle tree, which is a type of data structure that allows for efficient verification of the contents of a large data set. The root of the Merkle tree is then used as the content address for

the entire file[73].When a user wants to retrieve a file, they can use its content address to find the closest peer in the network that has that file, and download it from there. IPFS uses a distributed hash table (DHT) to keep track of where files are stored, and a distributed network protocol called the Interplanetary Name System (IPNS) to allow for mutable content addressing[74].IPFS also supports other features like versioning, file pinning, and file encryption. It also allows to build decentralized apps and platforms with it, as well as using it to access content and data in decentralized way, with less dependency on centralized servers.[10].

2.5.2 The Features of IPFS

IPFS has a number of features that make it a powerful tool for creating decentralized systems:

- Content Addressing: Instead of using traditional URLs, IPFS uses content-addressed hashes to identify and retrieve files. This allows for efficient and secure distribution of large files[74].
- Decentralized: IPFS is a peer-to-peer network, which means that files are stored and retrieved directly from other users. This eliminates the need for central servers and makes the system more resilient to censorship and other forms of interference[75].
- Versioning: IPFS supports versioning of files, which allows for easy rollbacks and updates[76].
- File Pinning: IPFS allows users to "pin" files, which ensures that they are kept on the network and can be easily retrieved in the future[77].
- File Encryption: IPFS supports end-to-end encryption of files, which allows for secure sharing of sensitive data.[78]

- Interplanetary Name System (IPNS): IPNS allows for mutable content addressing, which means that users can update the content behind a specific address, while the address remains the same[79].
- Distributed Hash Table (DHT): IPFS uses a DHT to keep track of where files are stored and to help users find the closest peer that has the file they're looking for[80].
- File Caching: IPFS allows caching of files, so that if a file is requested multiple times, it will be retrieved from the local cache rather than being downloaded again[80].
- Interoperability: IPFS can work in conjunction with other peer-to-peer protocols and is compatible with existing web infrastructure[74].
- Built-in File Format Support: IPFS supports multiple file format and protocols out of the box, like HTTP and SFS[81].

These features make IPFS a powerful tool for building decentralized systems, and it's being used in a variety of applications, including file storage and sharing, distributed web applications, and distributed databases.

2.5.3 Distributed hash table (DHT)

In IPFS, a Distributed Hash Table (DHT) it used to keep track of where files are stored and to help users find the closest peer that has the file they're looking for. A DHT is a distributed data structure that maps unique keys to values, and it allows for efficient lookups and updates in a peer-to-peer network[82]. Figure 2.5

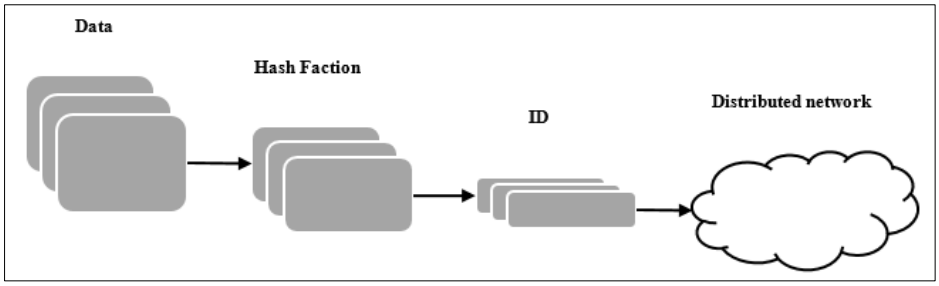


Figure 2.5 DHT Node Architecture[80]..

When a file is added to IPFS, it is broken into blocks and a content-addressable hash is generated for each block. The hash of the root of the Merkle tree is then used as the content address for the entire file. This content address is then used as the key in the DHT. When a user wants to retrieve a file, they use the content address to find the closest peer in the network that has that file[83]. The user first contacts a known peer in the network, called a "bootstrap node," which responds with a list of other peers that it knows about. The user then chooses a peer from this list and sends a message to it, asking for the file. The peer that receives the request first checks its local store to see if it has the file. If it does, it sends the file back to the user. If it doesn't, it sends a message to its closest known peer that has the file, and that peer sends the file back to the user. This process continues until the file is found, and each peer that is contacted adds the requesting peer to its list of known peers, so that future requests can be routed more efficiently. In summary, IPFS DHT is a distributed lookup mechanism that allows for efficient and secure retrieval of files in a peer-to-peer network. It allows for efficient lookups and updates in a peer-to-peer network, and it allows for efficient distribution of files and reduces the dependency on centralized servers[84].figure 2.6

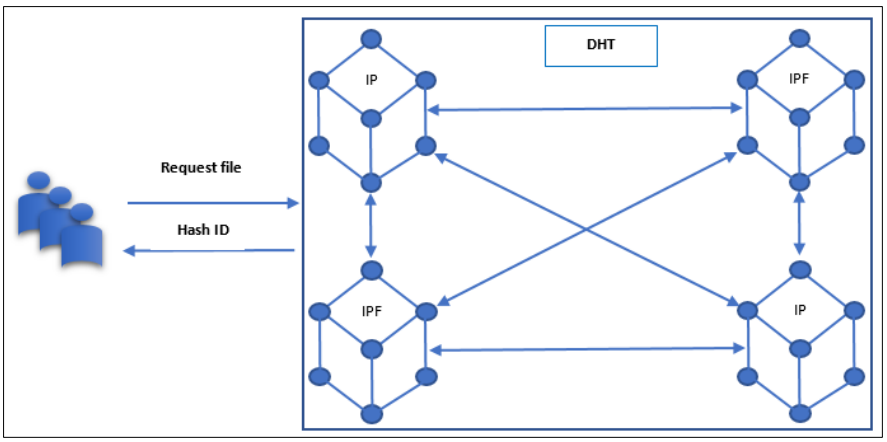


Figure 2.6 Design of IPFS based DHT[83]..

2.5.4 IPFS algorithm

IPFS (Interplanetary File System) uses a variety of algorithms to enable peer-to-peer file sharing and content addressing. Some of the algorithms used in IPFS include:

- SHA-256: IPFS uses the SHA-256 (Secure Hash Algorithm 256-bit) to hash all the content and generate a unique cryptographic hash for each file. This hash is used as the content's address on the IPFS network[7].
- Kademlia: IPFS uses the Kademlia algorithm for its distributed hash table (DHT) implementation. Kademlia is a distributed hash table algorithm that allows nodes to find the closest peers in the network that have a specific piece of content[85].
- SFS: IPFS uses SFS (Self-Certifying File System) to achieve decentralized naming, it allows IPFS nodes to use the IPFS network to host and access content, without the need for a central authority[86].

- BitSwap: IPFS uses BitSwap, a peer-to-peer file-trading algorithm that allows nodes to trade pieces of data with each other, based on the content they are looking for and the content they have to offer[87].
- MerkleDag: IPFS uses MerkleDag data structure to organize files into a directed acyclic graph (DAG). This allows for efficient content addressing and de-duplication of data[88].
- libp2p: IPFS uses libp2p, a peer-to-peer networking stack, to handle the peer-to-peer communications that underlie the IPFS network[6].

2.5.5 Application of IPFS in Blockchain Technology

IPFS (Interplanetary File System) can be used in conjunction with blockchain technology to create decentralized, peer-to-peer applications. Here are a few ways that IPFS and blockchain can be used together:

- Decentralized storage: IPFS can be used to store files on a decentralized network of nodes, while a blockchain can be used to record the location of these files and track who has access to them. This creates a tamper-proof and censorship-resistant method of storing data[88].
- Decentralized websites: IPFS can be used to host websites on a decentralized network, while a blockchain can be used to register and manage domain names. This creates a way to host websites that is not controlled by a central authority[88].
- Decentralized file sharing: IPFS can be used to share files peer-to-peer, while a blockchain can be used to track who has access to the files, and to facilitate payments for access to the files[75].

- Smart Contracts: IPFS can be used to store the data of smart contracts on a decentralized network, while blockchain can be used to execute these contracts and keep track of the state of the contracts on the network[89].
- IPFS and blockchain technologies complement each other well, IPFS provides decentralized storage and content addressing, while blockchain provides a secure and immutable ledger to record and track data. Together they can create decentralized applications that are resistant to censorship and tampering, while also providing a way to monetize the data and services provided by the application[89].

2.6 Security Objectives (CIA)

The CIA model outlines the three primary goals of cybersecurity. The letter C stands for confidentiality. Data and information privacy is essential for cybersecurity. Usernames, password combinations, medical histories, and other data, files, and staff must be permitted or confined to specific people, equipment, or procedures[90]. Though many issues might arise if the wrong people get access to information and data users really aren't supposed to have seen, confidentiality is focused on the accessing of data and information. In the CIA paradigm, the letter I stands for integrity. Users ought to be ensured that the information that is transferred analyzed and stored has still not been altered, either unintentionally or forcibly. For example, whenever a message is modified in one place, it can modify everywhere. The overall transmission could even be twisted or unintelligible[91]. It means that the final letter of the alphabet is available. Availability guarantees that users who are permitted to perform

their work can do so even with all of the cybersecurity precautions in place for working with equipment, software, individuals, procedures, and therefore more. Legitimate people should have immediate access to the tools they should be doing their work, and in the event of a cybersecurity incident or catastrophe, the system should be completely resilient and load-balanced[92].(see figure 2.8)

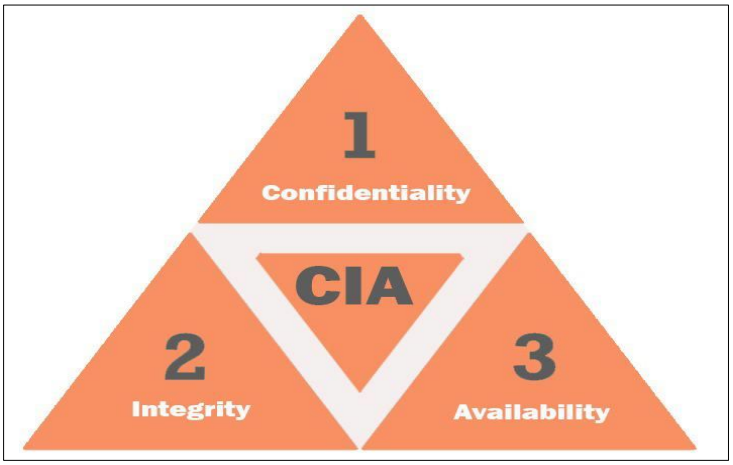


Figure 2.7 security objectives (CIA) [91].

2.6.1 The Main Classes of Cryptographic Algorithms

Cryptographic algorithms can be categorized into three main classes. This categorization is defined on basis of the number of cryptographic keys that are required for the algorithm.

- Hash Functions Hash functions are the building blocks for modern cryptography. A hash function is a cryptographic algorithm which is used to transform large random size data to small, fixed size data. The data output of the hash algorithm is called hash value or digest. The basic operation of hash functions does not need any key and operates

in a one-way manner. The one-way operation means that it is impossible to compute the input from a particular output [93] [94], The basic uses of hash functions are:

1. Generation and verification of digital signatures
2. Checksum/Message integrity checks
3. Source integrity services via Message Authentication Code
4. Derivation of sub-keys in key-establishment protocols and algorithms
5. Generation of pseudorandom numbers

- Symmetric-key algorithms, also referred as secret-key algorithms, use a single cryptographic key for encryption and decryption purposes. They convert data in a way that is problematic for an opponent to decrypt the data without the key. Symmetric keys are securely generated and distributed to the sender and receiver and are unknown to any other entity [95].If a symmetric-key algorithm is being used by more than one receiver, then the key has to be shared with all entities. If the key is compromised from one entity, communication of all the entities will be compromised. Symmetric Algorithms are further divided into Block & Stream algorithms [96].A block algorithm breaks the input into fixedsize blocks and then progresses the crypto operations. Stream algorithms perform “bit-by-bit” crypto operations. Primary purposes of symmetric key algorithms are:

  1. Confidentiality is achieved as encryption and decryption is performed using single key.

  2. Integrity and source authentication is achieved by using Message Authentication Codes because the Message Authentication Code is generated and validated by the same key.

  3. Generation of pseudorandom random numbers

- Asymmetric-key algorithms : are commonly referred to as “public-key algorithms”. They use two mathematically associated keys knows as public and private keys. One key is used for data encryption, and the other is used for decryption of data. The combination of a public and private key is called a key pair. The private key is always kept secret by the owner. The public key is distributed to the public and everyone can access it. The private key cannot be deduced from the public key [97]. The public key is mostly bound to an identity by a Certificate Authority. Asymmetric-key algorithms are mostly based on mathematical problems like integer factorization and discrete logarithm problem [98].Main uses of asymmetric algorithms are:
  1. Creation of digital signatures.
  2. To establish/distribute session keys.

2.6.2 RSA

Is a widely used public key cryptography algorithm that is commonly used for secure data transmission. It works by using two keys, a public key and a private key, to encrypt and decrypt data. The public key is used to encrypt the data and the private key is used to decrypt it.[99]
Here's a simple explanation of how RSA works:

- Key Generation: The first step is to generate the public and private keys. This is typically done by a trusted third-party, such as a certificate authority[100].

- Encryption: When sending a message, the sender uses the recipient's public key to encrypt the data. The encrypted message can only be decrypted using the recipient's private key.[101]

- Decryption: The recipient uses their private key to decrypt the encrypted message. Since the private key is kept confidential, only the intended recipient can read the message[101].
- Digital Signatures: RSA can also be used for digital signatures. A digital signature is a way to verify the authenticity and integrity of a message. To create a digital signature, the sender uses their private key to encrypt a hash of the message. The recipient then uses the sender's public key to decrypt the signature and verify the hash of the message.[102]

The RSA algorithm is implemented in general pseudo code below:

Algorithm 2.1: key generation

```
input: none
output: public_key, private_key
// Choose two large prime numbers p and q
    p, q = random_large_primes()
    // Compute the modulus n
    n = p * q
    // Compute the totient of n
    totient = (p - 1) * (q - 1)
    // Choose an integer e that is relatively prime to totient
    e = random_relative_prime(totient)
    // Compute the modular multiplicative inverse of e
    d = modular_multiplicative_inverse(e, totient)
    // Return the public and private keys
    public_key = (e, n)
    private_key = (d, n)
    return public_key, private_key
    end function
```

Algorithm 2.2: Encryption

```
input: message, public_key
output: encrypted_message
    // Convert the message to an integer m
    m = integer_from_message(message)
    // Extract the modulus and exponent from the public key
    e, n = public_key
    // Compute the ciphertext c
    c = modular_exponentiation(m, e, n)
    // Convert the ciphertext to a string
    encrypted_message = string_from_integer(c)
    // Return the encrypted message
    return encrypted_message
    end function
```

Algorithm 2.3: Decryption

```
input: encrypted_message, private_key
output: message
    // Convert the encrypted message to an integer c
    c = integer_from_string(encrypted_message)
    // Extract the modulus and exponent from the private key
    d, n = private_key
    // Compute the plaintext m
    m = modular_exponentiation(c, d, n)
    // Convert the plaintext to the original message
    message = message_from_integer(m)
    // Return the decrypted message
    return message
    end function
```

2.6.3 Elliptic Curve Cryptography (ECC)

Elliptic Curve Cryptography (ECC) is a type of public key cryptography that uses the mathematics of elliptic curves to provide strong security. It is commonly used for secure communication over the internet, including SSL/TLS encryption for websites and encrypted messaging.[103]

Here's a simple explanation of how ECC works:

- Key Generation: The first step is to generate the public and private keys. This is typically done by the user, and the private key is kept confidential. The public key is used to encrypt the data and the private key is used to decrypt it.[104]
- Encryption: When sending a message, the sender uses the recipient's public key to encrypt the data. The encrypted message can only be decrypted using the recipient's private key.[105]
- Decryption: The recipient uses their private key to decrypt the encrypted message. Since the private key is kept confidential, only the intended recipient can read the message[105].
- Digital Signatures: ECC can also be used for digital signatures. A digital signature is a way to verify the authenticity and integrity of a message. To create a digital signature, the sender uses their private key to encrypt a hash of the message. The recipient then uses the sender's public key to decrypt the signature and verify the hash of the message[103].

Algorithm 2.4: key generation

input: none

output: private key, public key // Choose a random private key

private key = random_number_between(1, curve_order)

// Compute the corresponding public key

public key = point multiply(private key, base_point)

end function

Algorithm 2.5: Encryption

input: message, recipient_public_key

output: encrypted_point, ephemeral_public_key

// Choose a random number k

k = random_number_between(1, curve_order)

// Compute the shared secret point

shared_secret = point_multiply(k, recipient_public_key)

// Compute the x-coordinate of the shared secret point

shared_secret_x = x_coordinate(shared_secret)

// Compute the hash of the shared secret x-coordinate

hash = hash_function(shared_secret_x)

// Convert the message to a point on the curve

message_point = point_from_message(message)

// Add the hash of the shared secret x-coordinate to the message point

encrypted_point = point_add(message_point, point_multiply(hash, base_point))

// Return the encrypted point and the ephemeral public key

ephemeral_public_key = point_multiply(k, base_point)

return encrypted_point, ephemeral_public_key

end function

Algorithm 2.6: Decryption

input: encrypted_point, recipient_private_key
output: message
// Compute the shared secret point
shared_secret = point_multiply(recipient_private_key, encrypted_point[1])
// Compute the x-coordinate of the shared secret point
shared_secret_x = x_coordinate(shared_secret)
// Compute the hash of the shared secret x-coordinate
hash = hash_function(shared_secret_x)
// Subtract the hash of the shared secret x-coordinate from the encrypted point
decrypted_point = point_subtract(encrypted_point[0], point_multiply(hash, base_point))
// Convert the decrypted point to the original message
message = message_from_point(decrypted_point)
// Return the decrypted message
return message
end function

In summary, ECC provides a secure way to transmit data over the internet by using a combination of public and private keys to encrypt and decrypt the data. ECC is more efficient than other public key cryptography systems, such as RSA, and provides equivalent security with smaller key sizes, making it well suited for use on resource-constrained devices, such as smart cards and mobile devices.

2.7 Ethereum platform

Ethereum is a public, open-source platform that runs on Blockchain technology and with smart contract functionality features. Ethereum is built as a Turing-complete scripting language. This is important due it needs to understand the agreements that allow for defining smart contracts. Ethereum is a programmable Blockchain. It allows anyone to build decentralized applications. Developers can use it to write code that controls digital assets and build any kind of application and not limited to crypto-currencies [106].Ethereum owns a digital crypto-currency called Ether (ETH). ETH, like Bitcoin, has many of the same features. It can be sent through the internet immediately. ETH is uncontrolled by the government or company (it is decentralized). Users around the world use it to pay for services on the network [76].

2.7.1 Node.je environment

Node.js is an open-source server environment that allows running JavaScript on the server; uses asynchronous programming to eliminate waiting and simply move on to the next request; runs single-threaded and non-blocking, and is very memory efficient. A common task for a web server is to open a file on the server and return the content to the client[107].

2.8 Truffle Suite framework

Truffle is a development environment, testing framework, and asset pipeline for Ethereum. It's a suite of tools that help developers build, test, and deploy decentralized applications on the Ethereum blockchain. Truffle includes the Truffle Framework, Truffle Develop, and Truffle Gas Station Network. The tools aim to make it easier for developers to build

and manage smart contracts, automate contract testing and deployment, and interact with the Ethereum network[108].

2.9 Web 3.js library

Web3 is a term used to describe the next generation of the internet, where data, applications, and services are decentralized, meaning they are not controlled by a single entity. The main idea behind web3 is to give users more control and ownership of their online data and information, instead of relying on centralized entities like corporations or governments. In the context of blockchain technology, web3 refers to the integration of decentralized applications (dapps) into the existing web infrastructure. This is achieved through the use of decentralized protocols and platforms, such as Ethereum, which provide the building blocks for web3 applications[109].

2.10 Meta Mask

MetaMask is a digital wallet that allows users to securely store, manage, and interact with various cryptocurrencies and decentralized applications. In simple terms, a tool helps people securely interact with blockchain technology[110].

2.11 Solidity Language

Solidity is a high-level, contract-oriented programming language for creating smart contracts on the Ethereum network. It has a syntax that is similar to C++, Python, and JavaScript because it is influenced by these languages. To ensure that smart contract code runs smoothly on the Ethereum Virtual Machine (EVM), Solidity is developed to complement EVM. Solidity smart contracts are saved and copied on the Ethereum blockchain network, outlining the terms and conditions of an agreement

between participants. Token sales, voting systems, and other decentralized apps can all have their behaviors defined in advance with Solidity programming[107].

2.12 Ganache blockchain

Ganache is a personal Blockchain for the Ethereum platform. It can be considered an Ethereum client. That allows us to run the test locally, deploy smart contracts, and develop an application in a secure and deterministic environment without needing to connect to a real Blockchain. Ganache has two versions: Command-Line Interface (CLI) as a command-line tool and Graphical User Interface (GUI) [111].

2.13 Performance Evaluation

This section presents the main interesting metrics and the dataset that employed to evaluate the proposed system. The Performance Evaluation is critical in checking the completed results for any study or research results. Therefore, choosing the right dataset and metrics is an essential key to differentiating in all performance evaluations**.**

2.13.1 Performance Metrics

In this study, the performance evaluation will be done using the same performance metrics which were employed by other studies in the related works. Thus, to be used multiple different metrics for validating the model of this thesis, such as cost and immutability, while the evaluation metrics are Recovery time, time estimated, Data storage and timely execution, giving us entire facts of how the proposed system will work.

2.13.2 Cost

A smart contract's cost is the amount of Crypto currency required to complete a transaction. To clarify the cost evaluation of the smart contract, it is necessary to refer to some important concepts[76]:

Gas: It is a unit that measures the amount of crypto currency required to execute each operation on Ethereum, as each operation on Ethereum, whether a smart contract instruction or a transaction, requires a certain amount of gas.

Gas Limit: The maximum amount of gas for all transactions created in the remix. Transaction Cost: The costs of transactions sent to the Ethereum Blockchain are determined by the contract's size.

Execution Cost: based on the cost of calculation operations that are executed as a result of the transaction.

2.13.3 Data Storage

Storing data on a blockchain can have a significant impact on the amount of storage required by the blockchain network. This is because each block in the blockchain contains a certain amount of data, and as more blocks are added to the chain, the amount of data stored on the network increases[61]..

- The amount of data stored on a blockchain can be considered as a metric, as it provides insight into the size and growth of the network, and can help to inform decisions around network scalability and resource allocation. However, it is important to note that the amount of data stored on a blockchain is not the only factor that affects network performance and scalability. Other factors, such as transaction volume, block size, and

network bandwidth, can also have a significant impact on network performance[64].

In addition, it's worth noting that there are limits to the amount of data that can be stored on a blockchain. For example, the Bitcoin blockchain has a maximum block size of 1 MB, which limits the amount of data that can be stored in each block. Other blockchain networks have implemented different approaches to scaling and data storage, such as sharding, sidechains, or off-chain storage solutions like IPFS, which can help to mitigate the impact of large amounts of data on the blockchain network.

Overall, while data storage is an important metric to consider in blockchain networks, it should be considered in conjunction with other metrics and factors that affect network performance and scalability.

2.13.4 Immutability

The concept of immutability in the context of blockchain refers to the property that once data is added to the blockchain, it cannot be altered or deleted. This property is a fundamental characteristic of blockchain technology and is achieved through the use of cryptographic hashing and consensus mechanisms.

To compute the immutability results of a blockchain, you can follow these steps:

Step 1: Determine the hash algorithm used by the blockchain: Different blockchain platforms use different hashing algorithms, such as SHA-256 or Keccak-256. You will need to determine the specific algorithm used by the blockchain in order to compute the immutability results[5].

Step2: Identify the data that is being stored on the blockchain: Depending on the blockchain platform and application, different types of data may

be stored on the blockchain. For example, Bitcoin stores transaction data, while Ethereum can store smart contracts and associated data[5].

Step3: Compute the hash of the data: Using the hash algorithm identified in step 1, compute the hash of the data that is being stored on the blockchain. This will result in a unique hash value that represents the data[59].

Step4: Verify the hash: Once the data has been added to the blockchain, you can verify its immutability by recomputing the hash of the data and comparing it to the hash that is originally stored on the blockchain. If the two hashes match, then the data has not been altered or tampered with, and is considered immutable[59].

In general, the immutability of a blockchain can be considered a qualitative property rather than a quantitative one. That is, it is not typically expressed as a numerical value or score, but rather as a binary property (i.e. data is either immutable or it is not). However, there are some metrics that can be used to assess the security and robustness of a blockchain's immutability, such as the amount of computational power required to alter the blockchain's history or the length of time that must elapse before a block can be considered immutable [59].

2.14 Summary

This chapter explains the fundamental concepts of the techniques used in the proposed system, along with an explanation of the tools that are used for the practical implementation of this system and a discussion of their most crucial functions. In addition, work environment encryption algorithms have been explained.

CHAPTER 3

RESEARCH METHODOLOGY AND PROPOSED SYSTEM

3.1 Overview

This chapter goes over the entire process of storing and retrieving medical health files using two methods that make use of the proposed decentralized system. The system addresses the primary security issues associated with medical health files, such as authentication and authorization, and achieves CIA, which stands for confidentiality, data integrity, and data availability.

3.2 The environment of Proposed System

The proposed system provides a secure environment for patient medical files and other sensitive information to be stored and retrieved when needed using a combination of blockchain technology and distributed file systems (IPFS). As a result, we created an integrated environment to protect these files using a variety of tools .We'll discuss inmore detail these tools and how they function above as shown in (Figure 3.1)

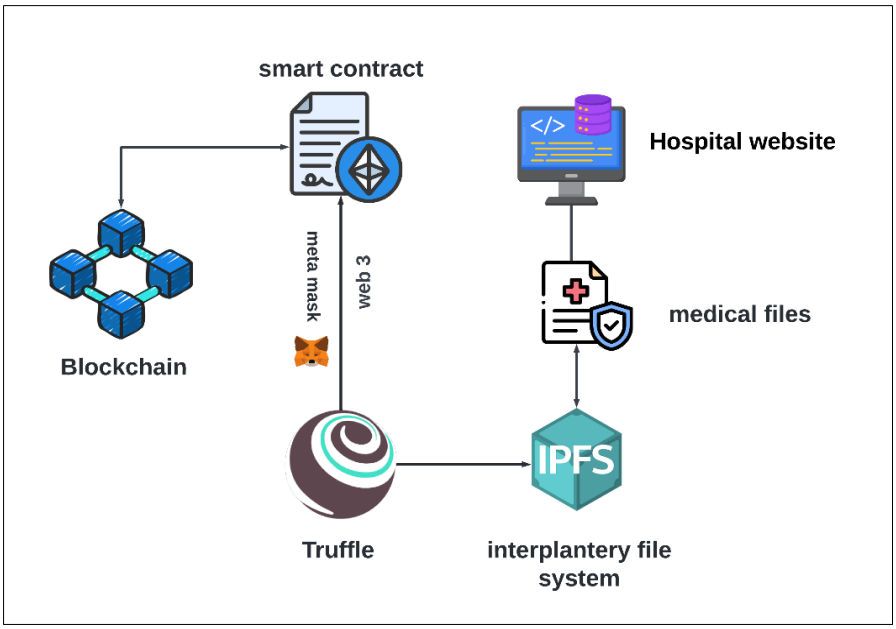


Figure 3.1 The Environmental Parts of the Proposed System.

3.2.1 Blockchain

A blockchain is a distributed system that uses cryptography to ensure the integrity of all transactions and blocks in the system. Each block in the chain provides as both an identity unit keeping its own information and a dependent link in the collective chain; this duality produces a network managed by the participants who store and distribute the information, as opposed to a third party. The blockchain used in this proposed system is a decentralized system that facilitates the use of applications built for the blockchain to store patient information. The root chunk's hash is paired with the decryption key to read each individual medical file. The information is protected from all save those who know the root chunk reference. As a result, the root pieces are locked up in immutable smart contracts on the blockchain and can be unlocked only under specific conditions.

3.2.2 Smart Contracts

The lines of code that make up a smart contract are designed to be activated and run automatically whenever they detect a specific action. The created smart contracts are pre-loaded with cryptographic keys that provide them the ability to encrypt any files that are generated as a result of an action being activated. A smart contract's primary purpose is to recognize the actions taken in relation to the data that sent and to inform the data into the system in order to determine whether the original data has been altered or not. The owner of the smart contract has the ability to decide whether or not an authorized third party may access the data that is requested by the owner. In the event that there is any sort of action taken on the data, the system will record the time at which the data last updated.

3.2.3 Truffle Suite

Truffle is an Ethereum-based testing framework and development environment that aims to simplify the developer experience. work easier. A truffle is a tool that allows users to build, compile, deploy, and test decentralized applications (DApps) on the blockchain.

3.2.4 IPFS

Known as Interplanetary File System When a file is added to IPFS, a hash value is generated; this protocol is meant to establish a group of peers that are all linked to the same (shared) filesystem, allowing for decentralized storage and multimedia sharing. Every node in the network has access to the file system and can upload and download files. Storage in IPFS is done via content addressing. This means that the contents themselves are used as identifiers in IPFS, rather than the location of the data (through, for example, an object's link).

3.2.5 Hospital Website

In the proposed system, we created a website for the hospital where users (patients, doctors, and other hospital staff) The hospital website, developed using IPFS and blockchain technology, provides a secure and user-friendly platform for hospital staff to securely store and access personal files. The decentralized approach ensures file availability and durability, while the blockchain component provides an immutable audit trail for transactions. Users can securely upload and store sensitive information, such as medical records, lab results, and other documents, with encrypted and protected files. The intuitive design and user-friendly features make file management easy, enhancing efficiency and patient care. The hospital website contributes to the overall efficiency,

collaboration, and quality of healthcare services within the hospital environment.

3.3 The Mechanism of the Proposed System

The basic idea is in the working mechanism of this proposed system, which in turn is divided into two processes, called the storage process and the retrieval process. Through these two processes, an integrated system will be formed to ensure the safety of patient records through the safe and efficient storage and retrieval methods described below in figure 3.2

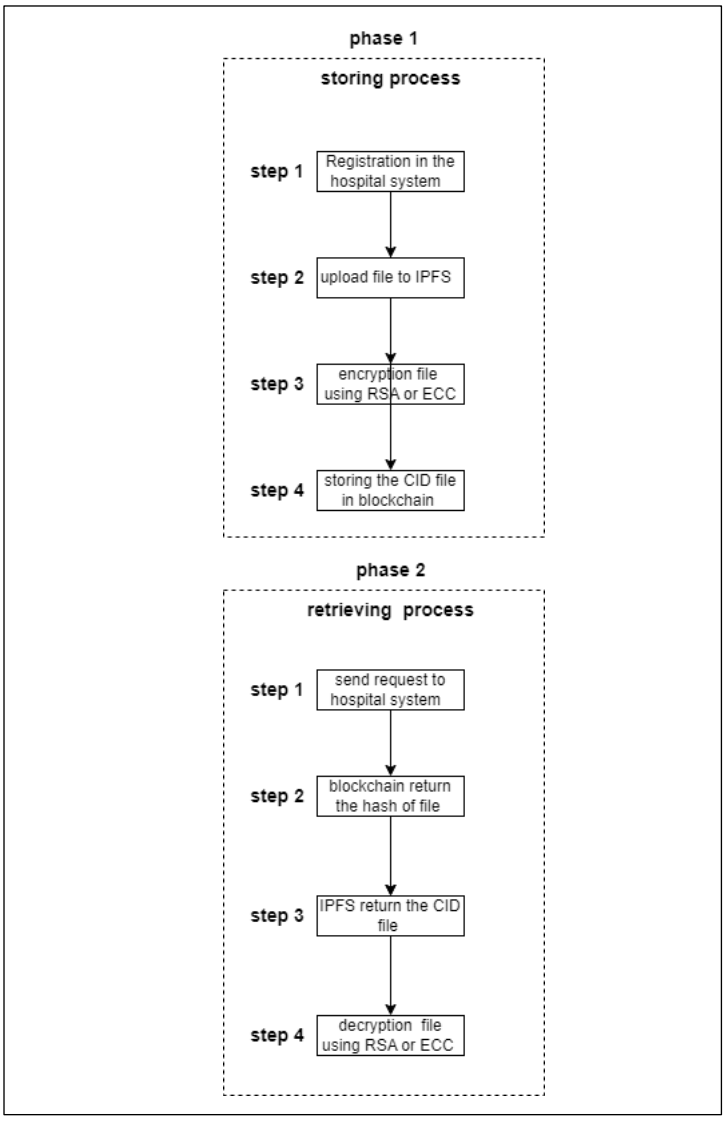


Figure 3.2 storage and retrieval file step

3.3.1 Storing process

The storage process consists of six steps:

1. The patient registers in the hospital system with all of his information (disease history, medical prescriptions, x-rays, etc.), allowing the doctor or hospital staff to create a file containing this patient's information.
2. The patient must have an account on the blockchain that is added with the patient's information.
3. For each patient, file, and doctor, the system generates a public and private key that the system uses to encrypt the data.
4. After the system generates the public and private keys, The patient's medical file is encrypted using the doctor's and patient public key with one of the proposed system's algorithms (RSA or ECC), and then uploaded to IPFS.

- [The aim of authentication and authorization of specific users within the system has been achieved. To ensure that these files are not modified by unauthorized access in the system].

5. The file is successfully uploaded to a local IPFS node. After the file is uploaded, IPFS uses one of the common hashing methods, such as SHA 256, SHA 3, and so on, to perform hashing and indexing, and each file is assigned a content identifier (CID).

- [Within the network, hashing is performed by a process of assigning keys to various values, and the resulting keys are then distributed to various nodes for later use in the retrieval process].

6. In this step, IPFS will return to smart contracts the content identifier (CID) of the file along with the hash. This enables the transaction to be sent and kept on the blockchain. Each

transaction must include the file CID and wallet address, as well as the patient's public key and the file's private key to be utilized later in the decryption process.

After completing the storing process, the system will have an encrypted file in IPFS with the content of a file (CID) and a hash stored within the blockchain. This means that confidentiality, integrity, and availability have all been achieved. As shown in block diagram Figure 3.2.

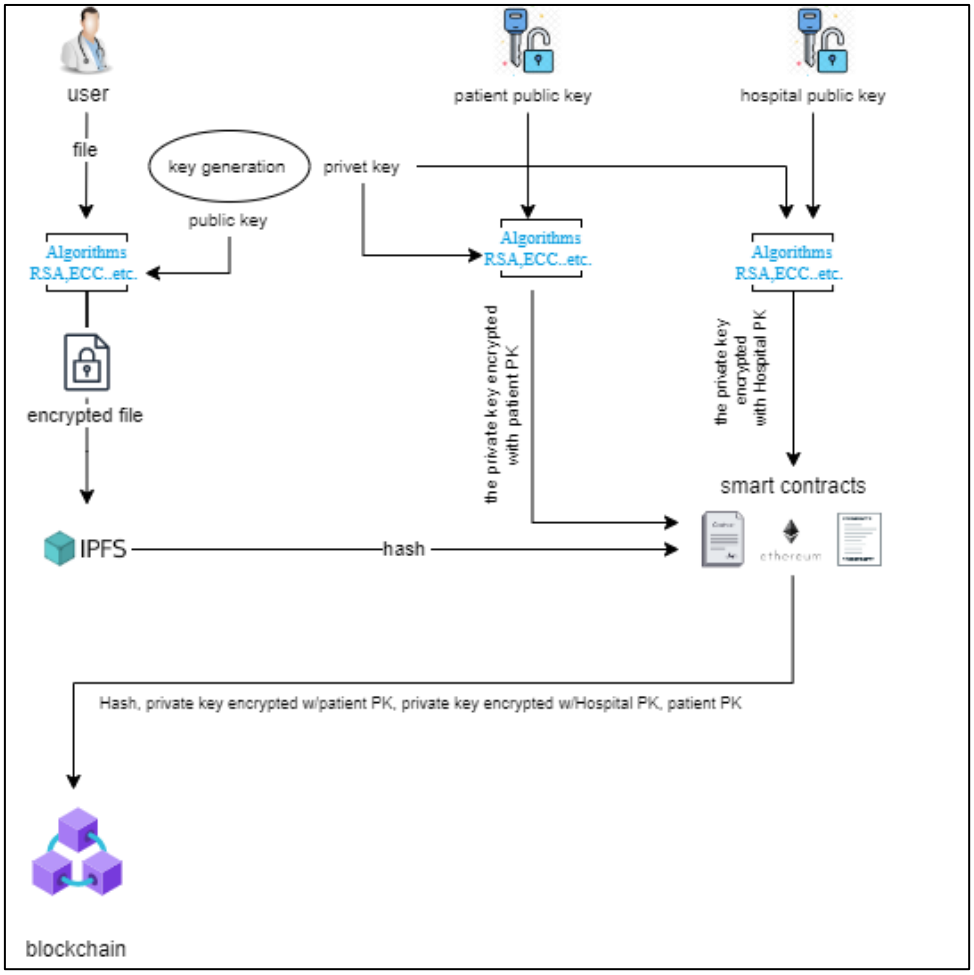


Figure 3.2 Storing Processes of the Medical Health Files.

3.3.2 The proposed system's storing process algorithm

Algorithm 3.1 :Storing process

```
FUNCTION medical_file_upload(mhf, patient_public_key,
hospital_public_key)
    // Input: mhf (medical health file), patient_public_key, hospital_public_key
    // Output: cid (file hash), decryption_keys, encrypted_mhf

    // Register patient in the hospital system
    patient_id = register_patient(patient_public_key, hospital_public_key)

    // Generate medical file for the patient
    mhf = generate_medical_file(patient_id)

    // Capture the file

    // Key generation - create a new pair of public-private keys
    decryption_keys = generate_decryption_keys()

    // Encrypt the file with patient's and hospital's public keys
    encrypted_mhf = encrypt_medical_file(mhf, patient_public_key,
hospital_public_key)

    // Encrypt the resulting pair's private key using the patient's public key
    encrypted_private_key = encrypt_private_key(decryption_keys.private_key,
patient_public_key)

    // Upload the encrypted file to the IPFS node
    cid = upload_file_to_ipfs(encrypted_mhf)

    // IPFS returns the Content ID of the encrypted file
```

```
    // Save the file to blockchain Content ID and wallet address using the patient's
public key and the file's private key
    save_file_to_blockchain(cid, patient_public_key, decryption_keys.private_key)

    // Return the output values
    return cid, decryption_keys, encrypted_mhf
END FUNCTION
```

3.3.3 Retrieving process

The storage process consists of seven steps:

1. The patient requests his medical files by sending a request through his wallet to the hospital system in which he is previously registered.
2. The system uses smart contracts to send this request to the blockchain.
3. The blockchain returns the hash of the requested file to the user.
4. The user sends the file's CID to IPFS, which uses it to look up the file's hash in the distributed hash table (DHT) in the IPFS node.
5. The encrypted file is retrieved by IPFS and returned with the file's private key.
6. Thus, the system decrypts the file using the hospital's and the patient's private key encrypted with one of the proposed system's algorithms (RSA or ECC).
7. All of the information in the file is visible after it has been decrypted. Additionally, complete privacy and security are maintained throughout the file retrieval process. As shown in block diagram Figure 3.3.

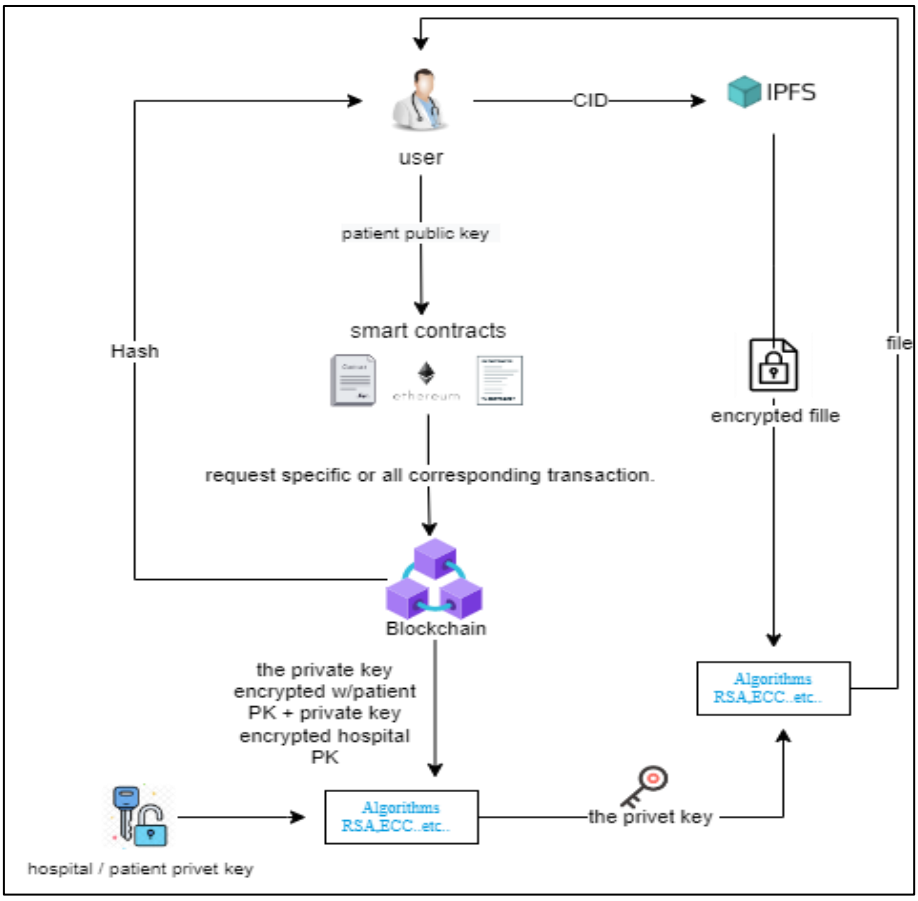


Figure 3.3 Retrieving Processes of the Medical Health Files.

3.3.4 The proposed system's retrieving process algorithm.

Algorithm 3.2: retrieving process

```
FUNCTION retrieve_file(patient_wallet, CID, private_key)
    // Input: patient_wallet (string), CID (string), private_key (string)
    // Output: file (string)

    // The patient requests his medical files from the hospital system
    request = make_request(patient_wallet, CID, private_key)

    // Smart contracts send this request to the blockchain.
    response = send_to_blockchain(request)
```

```
    // The blockchain returns the hash of the requested file.
    file_hash = extract_file_hash(response)

    // The user sends the file's CID to IPFS.
    encrypted_file = retrieve_from_IPFS(CID)

    // The encrypted file is retrieved by IPFS and returned with the file's private key.
    decrypted_file = decrypt_file(encrypted_file, private_key)

    // The system decrypts the file using the hospital's and the patient's private keys.
    file = decrypt_with_hospital_key(decrypted_file)

    // Return the file
    return file
END FUNCTION
```

3.4 Summary

In order to achieve the aims of privacy, integrity, and availability of the patient's medical health files, the process of storing medical health files and how to deal with them in complete privacy with the possibility of retrieval after encrypting them using encryption algorithms in the proposed system.

CHAPTER 4

IMPLEMENTATION, RESULTS, AND EVALUATION

4.1 Overview

This chapter presents an implementation and a discussion of the practical results of the proposed system to improve is secured and does efficient management of data privacy. It is important to note that the obtained results verify that our system of storing and retrieving medical health files is truly effectively appropriate to such privacy preservation of healthcare Data using blockchain technology and IPFS.

4.2 Implementation system requirements

The proposed system is implemented using a Lenovo laptop with the following specifications:

- Windows edition: Win 10
- The processor: Core i7 H 10th
- GPU: 4 GB GTX
- The memory (RAM): 16 GB
- System type: 64 bit

4.3 Tools of the Implementation

Among the most important tools that were used to implement this system are:

4.3.1 React, CSS, & JavaScript:

React.js and IPFS are two technologies that can be used to build decentralized medical health file applications. IPFS provides a distributed file system for storing and sharing medical files, while React.js provides a flexible and efficient front-end development framework. By using IPFS with React.js, developers can create web applications that are decentralized, fast, and secure for managing

medical health files. To integrate IPFS with React.js, developers can use the IPFS API to interact with the IPFS network from within a React.js application. Overall, React.js and IPFS provide a robust framework for building decentralized medical health file applications that offer a high level of security, privacy, and reliability. Developers can create web applications that provide patients with full control over their medical health data, enabling them to share their medical data securely with healthcare providers and organizations.

CSS is used to control presentation, formatting, and layout.

JavaScrip is used to control the behavior of different elements.

4.3.2 Node.js:

Node.js is an open-source server environment that allows you to run JavaScript on the server; uses asynchronous programming to eliminate waiting and simply move on to the next request; runs single-threaded and non-blocking, and is very memory efficient. A common task for a web server is to open a file on the server and return the content to the client.

Node.js is a program that can be downloaded from the website as in Figure 4.1A, and when it is installed on the PC. It contains all the JavaScript libraries. Any developer who develops a function in JavaScript will upload it to this website.

4.3.3 Truffle:

Truffle is a development framework for Ethereum-based blockchain applications, and it can be used in conjunction with IPFS to build decentralized medical health file applications. Developers can use the IPFS API to interact with the IPFS network from within a Truffle-based smart contract. By using Truffle with IPFS, developers can create smart

contracts that manage medical health files securely and transparently on the Ethereum blockchain. This ensures that patients have full control over their medical health data and can share it securely with healthcare providers and organizations. Truffle and IPFS provide a robust framework for building decentralized medical health file applications that offer a high level of security, privacy, and reliability.

It has a set of injunctions, a set of which have been used:

Truffle compile: This injunction will be used in the case of creating new smart contracts for the purpose of creating an image of a contract in the formula json.

Truffle migrate: This injunction is used for the purpose of uploading smart contracts on the Ethereum network. This injunction must be implemented whenever we update the code inside the smart contracts. Delete the migrate and make a new one.

Truffle console: This injunction is used to open truffle development, where it gives a CMD console and from which I can do development.

Npm run start: This injunction is used to start the server and open the website (Proposed System).

4.3.4 Meta Mask

MetaMask is a global community of developers and designers dedicated to making the world a better place with blockchain technology. The mission is to democratize access to the decentralized web, and through this mission, to transform the internet and world economy to one that empowers individuals through interactions based on consent, privacy, and free association.

It is an electronic wallet through which the exchange process takes place, and it is a gateway to the applications of the blockchain. It can be

obtained as an extension with a browser or as an application on the mobile. It is a solution to the problem of trust between the two parties (user & blockchain network), and it is considered a safe way to connect to applications built with the blockchain, and it is considered a key safe and entry Security, a token wallet, and everything a person needs to manage digital assets. Where it can be linked with the server of the blockchain network after creating an account inside it via the private key and placing electronic money (Ethereum) inside the wallet to be used in the completion of the programming process. Where in every process of adding information (node) to the blockchain network or modifying it, money will be required, as well as when uploading the code, and in every case of testing the code, it will need money.

4.3.5 Ganache:

Ganache is part of the Truffle Suite ecosystem. It is a program that can be downloaded from the Internet and installed on a personal computer and works on Windows operating systems in addition to Mac and Linux. Where it contains IP and a port, which makes it a suitable work environment for the implementation of the project. It can also be linked to truffle through the file it owns (truffle -config.js) where IP is placed and the port in the program ganache. It can also be viewed quickly, to see the current status of all accounts, including their addresses, private keys, transactions, and balances. It can be used to deploy contracts, develop applications, and run tests with quite easily.

The other feature of choosing it is a free program, where when building the system, we need to perform several tests, and the cost will become large if a program that requires costs when testing is used, and we can wait until the smart contracts become free from defects to be deployed

by paying the costs. It is also characterized by the speed of completion of the process, as it is considered one of the fastest platforms built with blockchain technology, as there are no obstacles hindering the process inside.

4.4 Deploy Smart Contract to Ethereum Network

The proposed system used the Ganache Ethereum test network to deploy smart contracts, record results, and evaluate system performance based on them. The following steps are to deploy a smart contract on the Ganache Ethereum test network to validate the smart contract.

1- The following are the configuration of the Ethereum network.

- Host Name:127.0.0.1
- Port number: 7545
- Network Id: 5777
- Account Default Balance:100 Ether

2- Open the MetaMask wallet and submit the amount of Cryptocurrency required to deploy a smart contract over the network for each Patient

3- The next step is deploying a smart contract from Visual Studio Code to the Ganache Ethereum local test network

4- A smart contract address is created to make transactions and call smart contract functions on the Ganache Ethereum network

5- Git Editor has been used for deploying the healthcare contract. Executing the script runs this.

Truffle migrate –reset// Reset the Deployment of the smart contract

4.5 System implementation stages

There are some implementation stages including storing process and retrieving stages.

4.5.1 Implementation of Storing process

Step 1: Develop a truffle environment with a set of instructions to execute smart contracts (see Figures of all steps in appendix)

Step 2: After the ganache server is installed, the smart contracts are compiled and migrated, and an IP address and port are added to begin the file-storing procedure.

Step 3: is attempting to (MetaMask) and making a wallet where got the private key, finishing the procedures for making the wallet, and putting digital money (Ethereum) inside it to use while running tests or implementing the program, because every test process or implementation of any work inside the wallet requires the use of Ethereum. Program participants will be compensated for referring clients to this portfolio. The steps for making a wallet in MetaMask are depicted in figure 6 in appendix

Step 4: Initially, through NPM start instructions, the hospital website will be opened to upload the files to the IPFS after being encrypted with one of the algorithms available on our medical website after the wallet and the account have been created in Blockchain.

Step 5: After the file is uploaded to the website, it will be sent to the server (Ganache) after being transacted, where it will appear on the blocks page with the number of blocks (nodes) in the chain, as well as the date of creation of each node and the amount of Ethereum needed to create it. On the Transactions page, the account that added it and the public key that sent and received it will appear.

4.5.2 Implementation of Retrieving process

Step 1: After uploading the files to the hospital site, they are Stored in IPFS in a hash form (CID) for each file in order to retrieve them through it.

Step 2: To complete the retrieval procedure and display the file, the file is chosen, copied, and displayed**. (**see Figures of all steps in appendix)

4.6 Performance Evaluation of Proposed System

In this section, the proposed system is evaluated according to the metrics that have been adopted (see Section 2.19 in Chapter Two).

1- Time storing and retrieving Results

The time it takes to store files in IPFS with different encryption algorithms like RSA and ECC will depend on a variety of factors, including the size of the files, the complexity of the encryption algorithms, the processing power of the computer used to encrypt the files, and the network speed of the computer used to store the files on IPFS.

Generally speaking, encryption algorithms like RSA and ECC are computationally intensive, especially when used with large files. This

means that encrypting files with these algorithms may take more time than encrypting files with simpler encryption algorithms, like AES. However, the tradeoff is that RSA and ECC are generally considered to be more secure than AES and other symmetric encryption algorithms, especially for applications where the keys need to be distributed securely.

When it comes to storing encrypted files on IPFS, the encryption algorithm used should not significantly affect the time it takes to store the files, since the files are stored in their encrypted form regardless of which encryption algorithm is used. However, the size of the encrypted files may be larger than the original files, which could affect the time it takes to transfer the files over the network.

In general, the performance impact of using RSA and ECC with IPFS will depend on the specific use case and the resources available. If security is a top priority and the files are not too large, using RSA or ECC may be a good choice. However, if performance is more important than security, using a simpler encryption algorithm like AES may be a better choice.

We took different sizes of files with different types. When it comes to storing the encrypted files on IPFS, the size of the encrypted files may be larger than the original files, depending on the encryption algorithm used. For example, RSA encryption typically results in larger encrypted files than ECC, since RSA uses larger key sizes. This means that storing the encrypted files on IPFS may take more time and network resources than storing the unencrypted files as shown in Figure (4-1).

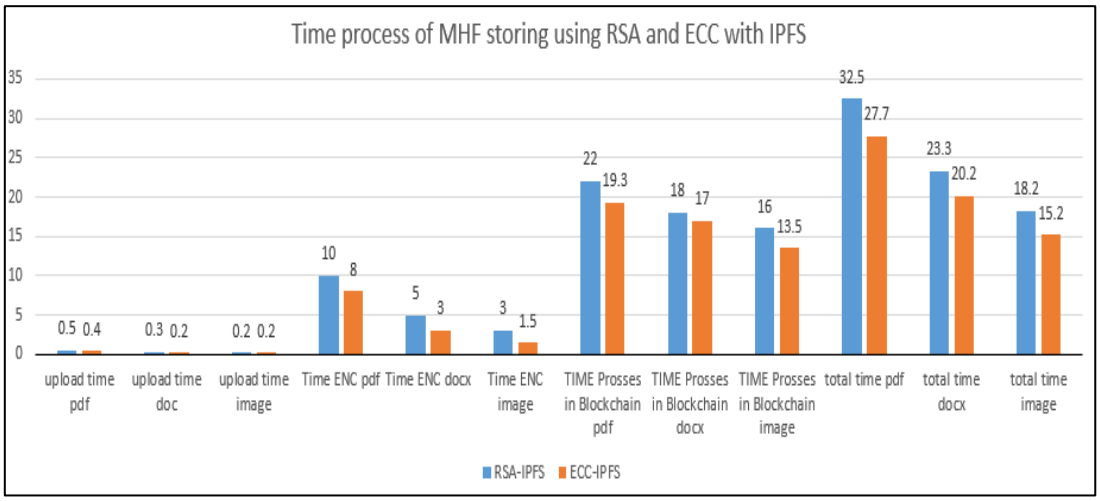


Figure (4-1) time of storing process with IPFS

Storing files directly on a blockchain can take significantly more time and resources than using IPFS. This is because blockchains are designed to be immutable, append-only ledgers that store transactional data, and are not optimized for storing large amounts of data, such as files.

When a file is stored on a blockchain, it needs to be divided into small pieces or "chunks", each of which is added to a block in the blockchain. Each block in the blockchain needs to be validated and verified by the nodes in the network, which requires significant computational power and time. As a result, storing large files directly on a blockchain can be prohibitively expensive and slow.

On the other hand, IPFS is a distributed file system that is specifically designed for storing and sharing large files. When a file is added to IPFS, it is divided into smaller chunks, which are then distributed across the IPFS network. This allows for faster and more efficient storage and retrieval of large files, without the need for expensive validation and verification processes.

If you were to store the output of IPFS on a blockchain, the process would still involve dividing the file into chunks, but these chunks would be added

to the blockchain as a reference to the original file stored on IPFS, rather than the actual file data. This approach is known as "off-chain" storage and is a common technique used in blockchain applications to store large files.

In summary, storing files directly on a blockchain can be slow and expensive, while using IPFS for file storage and referencing the IPFS content identifier (CID) on the blockchain is a more efficient and cost-effective approach as shown in figure (4-2).

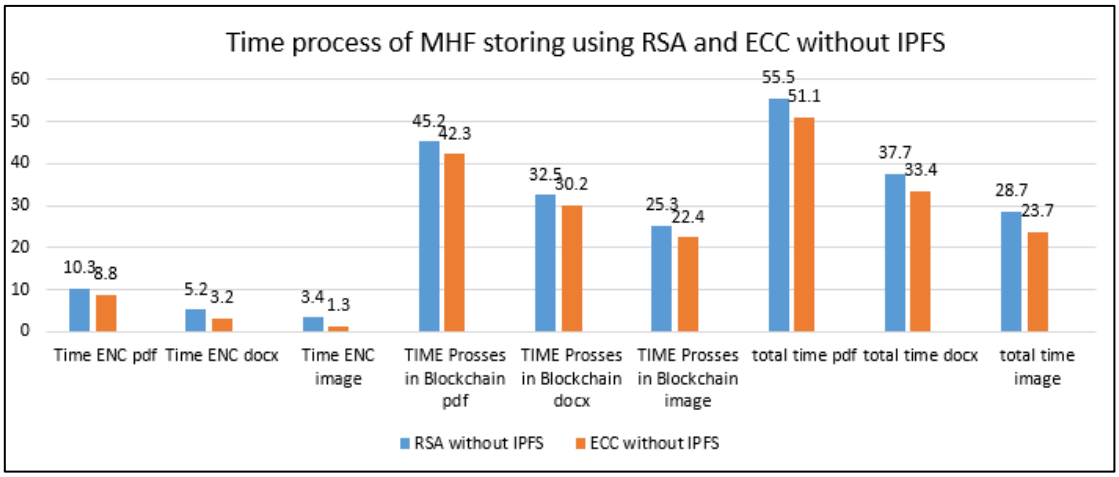


Figure (4-2) time of storing process without IPFS

Retrieving files from a blockchain can be more difficult and resource-intensive than retrieving files from IPFS, especially if the files are stored directly on the blockchain without using IPFS.

When a file is stored on a blockchain, it is typically broken up into smaller chunks or fragments, which are stored in separate blocks on the blockchain. Retrieving the original file requires reassembling these fragments, which can be time-consuming and require significant computational resources.

In contrast, retrieving files from IPFS is generally faster and more efficient, since IPFS uses content-addressable storage, which means that files are stored and retrieved based on their unique content identifier (CID). When a file is requested from IPFS, the CID is used to locate the file and retrieve it from the IPFS network.

In general, using IPFS to store and retrieve files can be faster and more efficient than using a blockchain, especially for large files. However, there may be situations where it makes sense to store files on a blockchain, such as when immutability and trust are important considerations, or when there are specific regulatory or compliance requirements that mandate the use of a blockchain. In these cases, it is important to carefully consider the trade-offs between using a blockchain and IPFS, and to design our system accordingly as shown in Figure (4-3) and (4-4).

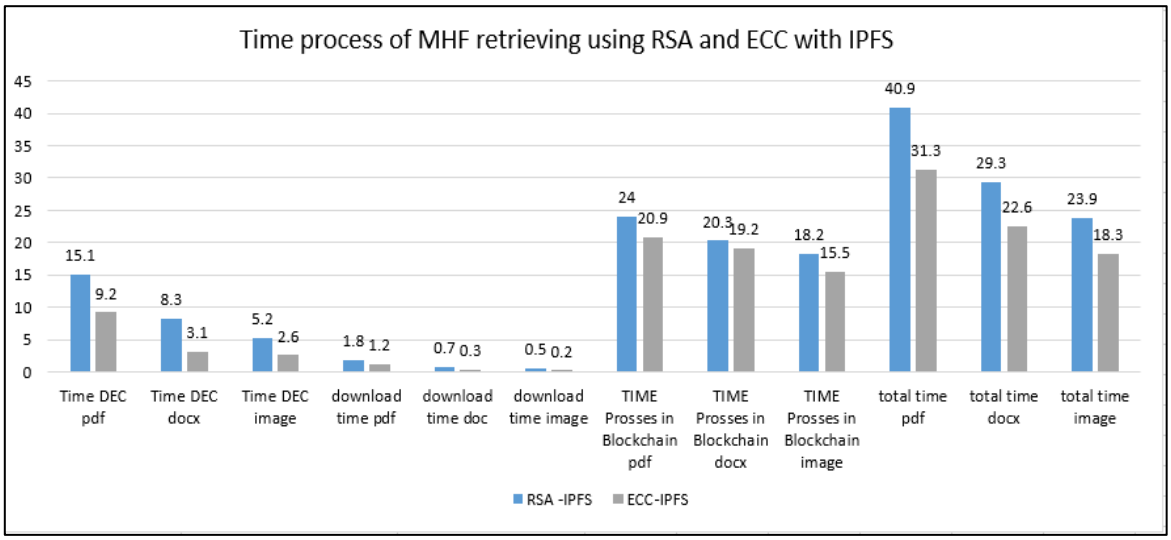


Figure (4-3) Time process of retrieving with IPFS

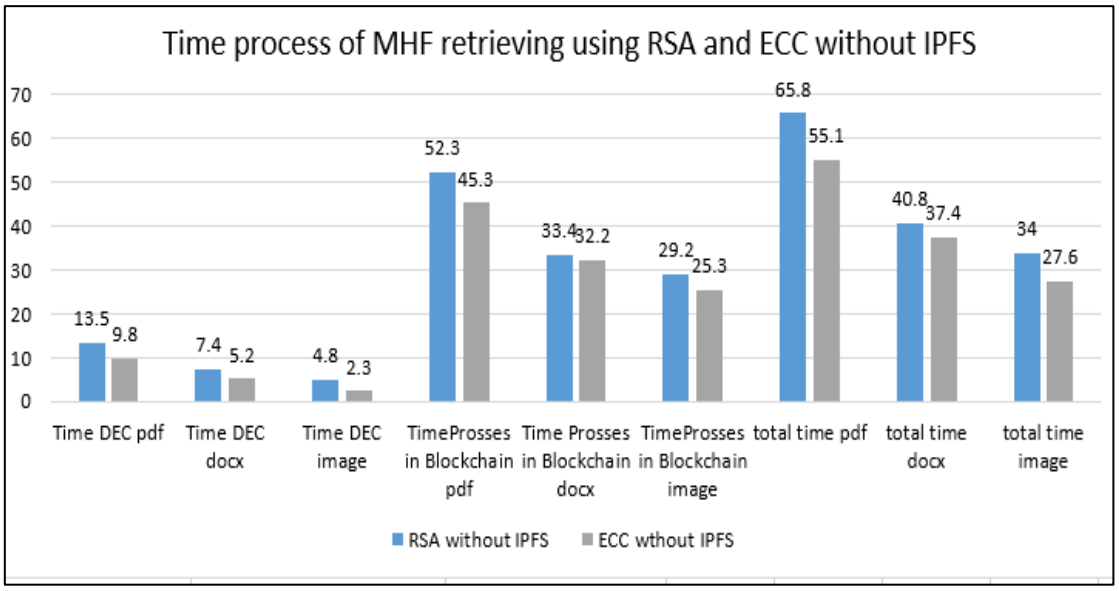


Figure (4-4) Time process of retrieving without IPFS

2- Execution Time Results

The execution time to deploy smart contracts and run their tests. We checked how well the proposed system worked by measuring how long it took this smart contract to run on average from the time it uploaded to the Ganache Network. This Blockchain simulator contains almost instant mining, which greatly reduces the time for testing execution. As a result, it found that the execution time of transmitting patient records without the framework proposed will take time 9.6 seconds. On the other hand, the execution time using the proposed framework, which took 2.6 seconds. In the end, we can show that the proposed smart contract-based model is possible because our model takes very little time and has very little overhead, which doesn't have a big effect on the Blockchain network or its users.

3- Cost

To compute the cost of storing data on the Ganache blockchain, we need to consider the gas cost. Gas is a unit of measurement used to represent

the computational effort required to execute a transaction or contract on the Ethereum blockchain, which is the underlying technology behind Ganache.

When we store data on the Ganache blockchain, we need to send a transaction to the network that includes the data we want to store. This transaction will require a certain amount of gas to be executed, and the cost of the gas is measured in Ether (ETH), which is the native cryptocurrency of the Ethereum network.

The cost of storing data on the Ganache blockchain will depend on several factors, including the size of the data being stored, the complexity of the transaction required to store the data, and the current gas price on the network.

To estimate the cost of storing data on the Ganache blockchain, we can use a tool like the Ethereum Gas Station (https://ethgasstation.info/) to look up the current gas price and estimate the gas cost based on the size of the data being stored and the complexity of the transaction.

For example, if we want to store a 1 KB file on the Ganache blockchain using a simple transaction, we might estimate a gas cost of around 100,000 units of gas. If the current gas price is 10 Gwei (0.00000001 ETH), then the cost of the transaction would be:

100,000 gas x 10 Gwei = 0.001 ETH

Keep in mind that the gas cost can fluctuate based on network congestion and other factors, so the actual cost may be higher or lower than our estimate.

4- Immutability Results

We have a blockchain that is used to store medical records. Each medical record is represented as a set of data fields such as patient name, date of birth, and medical history. The blockchain uses the SHA-256 hashing algorithm to compute the hash of each medical record and stores the hash values in blocks.

To compute the immutability results of this blockchain, we can follow these steps:

Step1: Determine the hash algorithm used by the blockchain: In this case, the blockchain uses the SHA-256 hashing algorithm.

Step2: Identify the data that is being stored on the blockchain: The blockchain is used to store medical records, which are represented as a set of data fields.

Step3: Compute the hash of the data: For each medical record, compute the SHA-256 hash of the data fields. For example, if a medical record contains the fields "Ali Ahmed", "01/01/1980", and "history of heart disease", then the hash of the data could be computed as follows:

SHA256("Ali Ahmed"+"01/01/1980"+"history of heart disease")

Step4: Verify the hash: Once the medical record has been added to the blockchain, the immutability can be verified by recomputing the hash of the data fields and comparing it to the hash that is originally stored on the blockchain. If the two hashes match, then the medical record has not been altered or tampered with, and is considered immutable.

In this example, the immutability of the medical records on the blockchain is achieved by using the SHA-256 hashing algorithm to compute a unique

hash for each record. The hash values are stored on the blockchain, and can be used to verify the integrity and immutability of the records.

5- Data Storage Results

In the case where IPFS is used to store files instead of storing them directly on the blockchain, the amount of data stored on the blockchain will be significantly reduced. This is because instead of storing the entire file on the blockchain, only the CID (Content Identifier) of the file is stored, which acts as a pointer to the file stored on the IPFS network.

By using IPFS, the blockchain network can reduce its storage requirements and improve its scalability, since it does not need to store large files directly on the blockchain. Instead, the IPFS network is responsible for storing and retrieving the actual file content.

However, it's important to note that using IPFS to store files does add some overhead to the process of retrieving files, since it requires additional steps to locate and retrieve the file from the IPFS network. This overhead can add some additional latency to the process of retrieving files, which may impact performance in some use cases.

Overall, the decision to use IPFS or not will depend on the specific requirements and constraints of the application or network. If storage requirements are a primary concern, then using IPFS can help to reduce the amount of data stored on the blockchain. However, if performance is a primary concern, then it may be better to store files directly on the blockchain, at the expense of increased storage requirements.

4.7 Comparing with Others Explorers

We made comparisons with other works in the same field and we considered several factors, such as methodology, results, limitations, and implication

Table 4.1 Comparing with Others Explorers

Ref. No.	Proposed solution	Technology used	Fully Decentralized	Security layer	Key exchange	Decentralized Access Control list
[1]	They proposed a model to reserve the authenticity, originality, and integrity of online books	IPFS and Ethereum smart contracts	✓	×	×	×
[2]	They a decentralized publication system to make the process of peer-review more transparent, faster and fair	IPFS and Ethereum Blockchain	✓	×	×	×
[3]	They proposed a decentralized system to securly sharing medical images	IPFS and Blockchain	×	✓	×	×
[4]	They proposed a system to decentralized, control and coordinate the he document version	IPFS and Ethereum smart contracts	✓	×	×	×
[5]	They proposed a secure system to store and share medical records	IPFS, Blockchain and Ethereum smart contracts	×	✓	✓	×
[6]	They proposed a combination of IPFS and Blockchain model to securely store medical records	IPFS and Blockchain	✓	✓	✓	×
[7]	They proposed a model that empower patients of manage their medical images.	IPFS and Ethereum Blockchain	✓	✓	✓	×

[8]	They proposed a patient centeric system to control their medical records	IPFS, Blockchain and Ethereum smart contracts	✗	✓	✓	✗
[9]	They proposed a distributed system to preserve the patient privacy by providing access to authorized authorities only.	IPFS and Blockchain	✓	✗	✗	✗
[10]	They proposed a system to control patient electronic records in a distributed manner	IPFS and Blockchain	✓	✓	✗	✗
Proposed System Of thesis	We proposed a decentralized system to manage medical files combined with access control list with each file	IPFS, Blockchain, Ethereum smart contract	✓	✓	✓	✓

CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1 Conclusions

The proposed system focuses on healthcare data privacy preservation and utilizes blockchain, IPFS, and cryptographic algorithms to achieve this goal. Various algorithms, methods, and methodologies are implemented to address privacy concerns. The system combines blockchain and IPFS to provide secure storage and retrieval solutions for sensitive data. Smart contracts on the blockchain enable access to private data, while the IPFS network facilitates data transfer for data mining operations. The system aims to ensure confidentiality, integrity, and availability of shared healthcare data while protecting privacy. The proposed strategies demonstrate usefulness in masking sensitive data and achieving decentralized healthcare data management. The approach emphasizes the prioritization of sensitive data protection and highlights the need for reducing side effects on non-sensitive data. The proposed system offers secure and efficient data privacy management with high storage capacity. It enables the creation of a private permissioned peer-to-peer blockchain network, addresses privacy risks, and ensures security and privacy through the combination of hash key, IPFS, blockchain, and cryptographic algorithms.Finally, we conclude that, when compared to current techniques, the proposed system allows for the creation of a private permissioned peer-to-peer blockchain network of multiple identifiable and registered stakeholders to achieve maximum interoperability, security, scalability, and per missioning. The system had solved the problem of third-party participation for data transfers to the data server using IPFS. The suggested strategy addresses the majority of privacy risks while taking into account the resource constraints of blockchain. Finally, based on provided methodologies, a healthcare data access control for total privacy records has been capable of guaranteeing security and privacy by

combining the benefits of the hash key, IPFS, blockchain, and cryptographic algorithms.

5.2 Limitation

In this section, some of the limitations are presented which had not been considered in the proposed system.

- The proposed system has not been considering the effect on the network's performance, Because of the focus on the storage and security sides.
- The security analysis is not considered.
- Difficulty in Implement the new build the smart contracts on the real Blockchain network, because of issues (cost and security).
- In addition, the proposed system has not been evaluated on multiple datasets of varying sizes and formats.

5.3 Future Works

This thesis is the first to address the challenge of providing comprehensive protection against both privacy and security risks in healthcare data, and it opens up numerous new research directions:

- Examine the link between blockchain and healthcare data privacy preservation using data anonymization approaches other than those examined in this thesis.
- Propose novel privacy-preserving procedures and strategies for healthcare data. This will necessitate a deeper dive into the legal literature on blockchain smart contracts.

- Improve the usefulness and performance of the present algorithms.
- Make the algorithms and analyses applicable to a variety of input data.
- Present real case studies in the context of healthcare data privacy and prevention.
- Extend concepts and methods to the analysis of healthcare data using blockchain in social network data.

Future studies will examine cutting-edge techniques for data processing and storage to provide qualities like fault tolerance or network resilience. By pushing the limits of healthcare data at the network's edge, smart, autonomous, embedded systems can support human activities while having less of an influence on the environment.

REFERENCES

[1] C. C. Agbo, Q. H. Mahmoud, and J. M. Eklund, "Blockchain technology in healthcare: a systematic review," in *Healthcare*, 2019, vol. 7, no. 2: MDPI, p. 56.

[2] M. Mettler, "Blockchain technology in healthcare: The revolution starts here," in *2016 IEEE 18th international conference on e-health networking, applications and services (Healthcom)*, 2016: IEEE, pp. 1-3.

[3] M. Hölbl, M. Kompara, A. Kamišalić, and L. Nemec Zlatolas, "A systematic review of the use of blockchain in healthcare," *Symmetry*, vol. 10, no. 10, p. 470, 2018.

[4] P. Zhang, D. C. Schmidt, J. White, and G. Lenz, "Blockchain technology use cases in healthcare," in *Advances in computers*, vol. 111: Elsevier, 2018, pp. 1-41.

[5] J. Jayabalan and N. Jeyanthi, "Scalable blockchain model using off-chain IPFS storage for healthcare data security and privacy," *Journal of Parallel and Distributed Computing*, vol. 164, pp. 152-167, 2022.

[6] R. Kumar, N. Marchang, and R. Tripathi, "Distributed off-chain storage of patient diagnostic reports in healthcare system using IPFS and blockchain," in *2020 International conference on communication systems & networks (COMSNETS)*, 2020: IEEE, pp. 1-5.

[7] S. Kumar, A. K. Bharti, and R. Amin, "Decentralized secure storage of medical records using Blockchain and IPFS: A comparative analysis with future directions," *Security and Privacy*, vol. 4, no. 5, p. e162, 2021.

[8] K. Azbeg, O. Ouchetto, and S. J. Andaloussi, "BlockMedCare: A healthcare system based on IoT, Blockchain and IPFS for data management security," *Egyptian Informatics Journal*, 2022.

[9] R. Kumar and R. Tripathi, "Towards design and implementation of security and privacy framework for internet of medical things (iomt) by leveraging blockchain and ipfs technology," *The Journal of Supercomputing*, vol. 77, no. 8, pp. 7916-7955, 2021.

[10] J. Sun, X. Yao, S. Wang, and Y. Wu, "Blockchain-based secure storage and access scheme for electronic medical records in IPFS," *IEEE Access*, vol. 8, pp. 59389-59401, 2020.

[11] M. M. Madine *et al.*, "Blockchain for giving patients control over their medical records," *IEEE Access*, vol. 8, pp. 193102-193115, 2020.

[12] G. Subramanian and A. S. Thampy, "Implementation of blockchain consortium to prioritize diabetes patients' healthcare in pandemic situations," *Ieee Access,* vol. 9, pp. 162459-162475, 2021.
[13] M. Barati, W. J. Buchanan, O. Lo, and O. Rana, "A Privacy-Preserving Platform for Recording COVID-19 Vaccine Passports," *arXiv preprint arXiv:2112.01815,* 2021.
[14] N. Rauta and K. Shah, "Implementation of Ethereum Blockchain in Healthcare Using IPFS," *Pulse,* vol. 2, no. 2, 2021.
[15] V. Mani, P. Manickam, Y. Alotaibi, S. Alghamdi, and O. I. Khalaf, "Hyperledger healthchain: patient-centric IPFS-based storage of health records," *Electronics,* vol. 10, no. 23, p. 3003, 2021.
[16] D. El Majdoubi, H. El Bakkali, and S. Sadki, "SmartMedChain: A Blockchain-Based Privacy-Preserving Smart Healthcare Framework," *Journal of Healthcare Engineering,* vol. 2021, 2021.
[17] M. M. Sheeraz, M. A. Islam, and H.-C. Kim, "A Decentralized Approach of Healthcare Data Collection for Research," in *INTERNATIONAL CONFERENCE ON FUTURE INFORMATION & COMMUNICATION ENGINEERING*, 2022, vol. 13, no. 1, pp. 143-148.
[18] A. Khatoon, "A blockchain-based smart contract system for healthcare management," *Electronics,* vol. 9, no. 1, p. 94, 2020.
[19] M. Y. Jabarulla and H.-N. Lee, "Blockchain-based distributed patient-centric image management system," *Applied Sciences,* vol. 11, no. 1, p. 196, 2020.
[20] M. Y. Jabarulla, G. Jung, and H.-N. Lee, "Decentralized Framework for Medical Images Based on Blockchain and Inter Planetary File System."
[21] R. K. Marangappanavar and M. Kiran, "Inter-planetary file system enabled blockchain solution for securing healthcare records," in *2020 third ISEA conference on security and privacy (ISEA-ISAP)*, 2020: IEEE, pp. 171-178.
[22] R. Kumar and R. Tripathi, "A Secure and Distributed Framework for sharing COVID-19 patient Reports using Consortium Blockchain and IPFS," in *2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, 2020: IEEE, pp. 231-236.
[23] A. Al Mamun, F. Jahangir, M. Umor, S. Azam, M. S. Kaiser, and A. Karim, "A combined framework of interplanetary file system and blockchain to securely manage electronic medical records," in *Proceedings of international conference on trends in computational and cognitive engineering*, 2021: Springer, pp. 501-511.

[24] S. Tanwar, K. Parekh, and R. Evans, "Blockchain-based electronic healthcare record system for healthcare 4.0 applications," *Journal of Information Security and Applications,* vol. 50, p. 102407, 2020.
[25] S. Shi, D. He, L. Li, N. Kumar, M. K. Khan, and K.-K. R. Choo, "Applications of blockchain in ensuring the security and privacy of electronic health record systems: A survey," *Computers & security,* vol. 97, p. 101966, 2020.
[26] C. Jarvis, *Physical examination and health assessment-Canadian E-book*. Elsevier Health Sciences, 2018.
[27] A. Roehrs, C. A. Da Costa, R. da Rosa Righi, and K. S. F. De Oliveira, "Personal health records: a systematic literature review," *Journal of medical Internet research,* vol. 19, no. 1, p. e5876, 2017.
[28] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi, "Deep EHR: a survey of recent advances in deep learning techniques for electronic health record (EHR) analysis," *IEEE journal of biomedical and health informatics,* vol. 22, no. 5, pp. 1589-1604, 2017.
[29] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren, "Secure, privacy-preserving and federated machine learning in medical imaging," *Nature Machine Intelligence,* vol. 2, no. 6, pp. 305-311, 2020.
[30] T. J. Hoffmann *et al.*, "A large electronic-health-record-based genome-wide study of serum lipids," *Nature genetics,* vol. 50, no. 3, pp. 401-413, 2018.
[31] A. Shahnaz, U. Qamar, and A. Khalid, "Using blockchain for electronic health records," *IEEE access,* vol. 7, pp. 147782-147795, 2019.
[32] C. Sowthily, S. Senthil Kumar, and M. Brindha, "Detection and classification of faults in photovoltaic system using random forest algorithm," in *Evolution in Computational Intelligence: Frontiers in Intelligent Computing: Theory and Applications (FICTA 2020), Volume 1*: Springer, 2020, pp. 765-773.
[33] A. G. Alexandru, I. M. Radu, and M.-L. Bizon, "Big Data in Healthcare-Opportunities and Challenges," *Informatica economica,* vol. 22, no. 2, 2018.
[34] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *2017 IEEE international congress on big data (BigData congress)*, 2017: Ieee, pp. 557-564.
[35] W. Nowiński and M. Kozma, "How can blockchain technology disrupt the existing business models?," *Entrepreneurial Business and Economics Review,* vol. 5, no. 3, pp. 173-188, 2017.

[36] A. A. Monrat, O. Schelén, and K. Andersson, "A survey of blockchain from the perspectives of applications, challenges, and opportunities," *IEEE Access*, vol. 7, pp. 117134-117151, 2019.
[37] A. Aswin, K. Basil, V. P. Viswan, B. Reji, and B. Kuriakose, "Design of AYUSH: A Blockchain-Based Health Record Management System," in *Inventive Communication and Computational Technologies: Proceedings of ICICCT 2019*, 2020: Springer, pp. 665-672.
[38] S. Wang, Y. Zhang, and Y. Zhang, "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems," *Ieee Access*, vol. 6, pp. 38437-38450, 2018.
[39] R. Xu, L. Zhang, H. Zhao, and Y. Peng, "Design of network media's digital rights management scheme based on blockchain technology," in *2017 IEEE 13th international symposium on autonomous decentralized system (ISADS)*, 2017: IEEE, pp. 128-133.
[40] A. Priya, A. Khatri, and P. Dixit, "Rise of blockchain technology: beyond cryptocurrency," in *Applications of Computing and Communication Technologies: First International Conference, ICACCT 2018, Delhi, India, March 9, 2018, Revised Selected Papers 1*, 2018: Springer, pp. 286-299.
[41] A. Angrish, B. Craver, M. Hasan, and B. Starly, "A case study for Blockchain in manufacturing:"FabRec": A prototype for peer-to-peer network of manufacturing nodes," *Procedia Manufacturing*, vol. 26, pp. 1180-1192, 2018.
[42] B. K. Mohanta, D. Jena, S. S. Panda, and S. Sobhanayak, "Blockchain technology: A survey on applications and security privacy challenges," *Internet of Things*, vol. 8, p. 100107, 2019.
[43] F. Calvão, "Crypto-miners: Digital labor and the power of blockchain technology," *Economic Anthropology*, vol. 6, no. 1, pp. 123-134, 2019.
[44] Y.-P. Chen and J.-C. Ko, "CryptoAR wallet: A blockchain cryptocurrency wallet application that uses augmented reality for on-chain user data display," in *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*, 2019, pp. 1-5.
[45] N. Chaudhry and M. M. Yousaf, "Consensus algorithms in blockchain: Comparative analysis, challenges and opportunities," in *2018 12th International Conference on Open Source Systems and Technologies (ICOSST)*, 2018: IEEE, pp. 54-63.
[46] A. Kuznetsov, I. Oleshko, V. Tymchenko, K. Lisitsky, M. Rodinko, and A. Kolhatin, "Performance Analysis of Cryptographic Hash

Functions Suitable for Use in Blockchain," *International Journal of Computer Network & Information Security*, vol. 13, no. 2, 2021.
[47] W. Wang *et al.*, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *Ieee Access*, vol. 7, pp. 22328-22370, 2019.
[48] P. Zhang and M. Zhou, "Security and trust in blockchains: Architecture, key technologies, and open issues," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 3, pp. 790-801, 2020.
[49] F. Casino, T. K. Dasaklis, and C. Patsakis, "A systematic literature review of blockchain-based applications: Current status, classification and open issues," *Telematics and informatics*, vol. 36, pp. 55-81, 2019.
[50] S. Seang and D. Torre, "Proof of Work and Proof of Stake consensus protocols: a blockchain application for local complementary currencies," *France: Universite Cote d'Azur-GREDEG-CNRS. Str*, vol. 3, no. 4, 2018.
[51] H. Xiong, M. Chen, C. Wu, Y. Zhao, and W. Yi, "Research on progress of blockchain consensus algorithm: a review on recent progress of blockchain consensus algorithms," *Future Internet*, vol. 14, no. 2, p. 47, 2022.
[52] P. Wagner, P. Birnstill, E. Krempel, S. Bretthauer, and J. Beyerer, "Privacy dashcam–towards lawful use of dashcams through enforcement of external anonymization," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology: ESORICS 2017 International Workshops, DPM 2017 and CBT 2017, Oslo, Norway, September 14-15, 2017, Proceedings*, 2017: Springer, pp. 183-201.
[53] P. Ekparinya, V. Gramoli, and G. Jourjon, "The attack of the clones against proof-of-authority. arXiv 2019," *arXiv preprint arXiv:1902.10244*, 2020.
[54] P. Dutta, T.-M. Choi, S. Somani, and R. Butala, "Blockchain technology in supply chain operations: Applications, challenges and research opportunities," *Transportation research part e: Logistics and transportation review*, vol. 142, p. 102067, 2020.
[55] E. Muminova, G. Honkeldiyeva, K. Kurpayanidi, S. Akhunova, and S. Hamdamova, "Features of introducing blockchain technology in digital economy developing conditions in Uzbekistan," in *E3S Web of Conferences*, 2020, vol. 159: EDP Sciences, p. 04023.
[56] M. Andoni *et al.*, "Blockchain technology in the energy sector: A systematic review of challenges and opportunities," *Renewable and sustainable energy reviews*, vol. 100, pp. 143-174, 2019.

[57] H. H. Khan, M. N. Malik, Z. Konečná, A. G. Chofreh, F. A. Goni, and J. J. Klemeš, "Blockchain technology for agricultural supply chains during the COVID-19 pandemic: Benefits and cleaner solutions," *Journal of Cleaner Production*, vol. 347, p. 131268, 2022.

[58] X. Xu, Z. Zeng, S. Yang, and H. Shao, "A novel blockchain framework for industrial IoT edge computing," *Sensors*, vol. 20, no. 7, p. 2061, 2020.

[59] M. Kizildag *et al.*, "Blockchain: A paradigm shift in business practices," *International Journal of Contemporary Hospitality Management*, vol. 32, no. 3, pp. 953-975, 2019.

[60] M. Miraz and M. Ali, "Applications of blockchain technology beyond cryptocurrency. arXiv 2018," *arXiv preprint arXiv:1801.03528*.

[61] B. K. Mohanta, S. S. Panda, and D. Jena, "An overview of smart contract and use cases in blockchain technology," in *2018 9th international conference on computing, communication and networking technologies (ICCCNT)*, 2018: IEEE, pp. 1-4.

[62] M. M. Queiroz, R. Telles, and S. H. Bonilla, "Blockchain and supply chain management integration: a systematic review of the literature," *Supply Chain Management: An International Journal*, vol. 25, no. 2, pp. 241-254, 2019.

[63] P. B. Marella, M. Milojkovic, J. Mohler, and G. G. Dagher, "GenVote: Blockchain-Based Customizable and Secure Voting Platform," in *Information Systems Security and Privacy: 4th International Conference, ICISSP 2018, Funchal-Madeira, Portugal, January 22-24, 2018, Revised Selected Papers 4*, 2019: Springer, pp. 152-171.

[64] A. Ghosh, S. Gupta, A. Dua, and N. Kumar, "Security of Cryptocurrencies in blockchain technology: State-of-art, challenges and future prospects," *Journal of Network and Computer Applications*, vol. 163, p. 102635, 2020.

[65] G. Malik, K. Parasrampuria, S. P. Reddy, and S. Shah, "Blockchain based identity verification model," in *2019 international conference on vision towards emerging trends in communication and networking (ViTECoN)*, 2019: IEEE, pp. 1-6.

[66] M. Takemiya and B. Vanieiev, "Sora identity: Secure, digital identity on the blockchain," in *2018 ieee 42nd annual computer software and applications conference (compsac)*, 2018, vol. 2: IEEE, pp. 582-587.

[67] T. M. Fernández-Caramés and P. Fraga-Lamas, "A Review on the Use of Blockchain for the Internet of Things," *Ieee Access*, vol. 6, pp. 32979-33001, 2018.

[68] T. Hewa, M. Ylianttila, and M. Liyanage, "Survey on blockchain based smart contracts: Applications, opportunities and challenges," *Journal of Network and Computer Applications,* vol. 177, p. 102857, 2021.
[69] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F.-Y. Wang, "Blockchain-enabled smart contracts: architecture, applications, and future trends," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 49, no. 11, pp. 2266-2277, 2019.
[70] L. Ante, "Smart contracts on the blockchain–A bibliometric analysis and review," *Telematics and Informatics,* vol. 57, p. 101519, 2021.
[71] M. Alharby and A. Van Moorsel, "Blockchain-based smart contracts: A systematic mapping study," *arXiv preprint arXiv:1710.06372,* 2017.
[72] Y. Chen, H. Li, K. Li, and J. Zhang, "An improved P2P file system scheme based on IPFS and Blockchain," in *2017 IEEE International Conference on Big Data (Big Data),* 2017: IEEE, pp. 2652-2657.
[73] E. Daniel and F. Tschorsch, "Ipfs and friends: A qualitative comparison of next generation peer-to-peer data networks," *IEEE Communications Surveys & Tutorials,* vol. 24, no. 1, pp. 31-52, 2022.
[74] J. Benet, "IPFS-content addressed, versioned, P2P file system (DRAFT 3)," *arXiv preprint arXiv:1407.3561,* pp. 1-11, 2014.
[75] T. V. Doan, V. Bajpai, Y. Psaras, and J. Ott, "Towards decentralised cloud storage with IPFS: Opportunities, challenges, and future directions," *arXiv preprint arXiv:2202.06315,* 2022.
[76] N. Nizamuddin, K. Salah, M. A. Azad, J. Arshad, and M. Rehman, "Decentralized document version control using ethereum blockchain and IPFS," *Computers & Electrical Engineering,* vol. 76, pp. 183-197, 2019.
[77] A. Mubashar *et al.*, "Storage and proximity management for centralized personal health records using an ipfs-based optimization algorithm," *Journal of Circuits, Systems and Computers,* vol. 31, no. 01, p. 2250010, 2022.
[78] H.-S. Huang, T.-S. Chang, and J.-Y. Wu, "A secure file sharing system based on IPFS and blockchain," in *Proceedings of the 2nd International Electronics Communication Conference,* 2020, pp. 96-100.
[79] S. Khatal, J. Rane, D. Patel, P. Patel, and Y. Busnel, "Fileshare: A blockchain and ipfs framework for secure file sharing and data provenance," in *Advances in Machine Learning and Computational*

Intelligence: Proceedings of ICMLCI 2019, 2021: Springer, pp. 825-833.

[80] Q. Zhang and Z. Zhao, "Distributed storage scheme for encryption speech data based on blockchain and IPFS," *The Journal of Supercomputing*, pp. 1-27, 2022.

[81] S. Routray and R. Ganiga, "Secure storage of electronic medical records (EMR) on interplanetary file system (IPFS) using cloud storage and blockchain ecosystem," in *2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 2021: IEEE, pp. 1-9.

[82] C. Bieri, "An Overview into the InterPlanetary File System (IPFS): Use Cases, Advantages, and Drawbacks," *Communication Systems XIV; University of Zurich: Zurich, Switzerland*, p. 78, 2021.

[83] L. Chen, X. Zhang, and Z. Sun, "Scalable Blockchain Storage Model Based on DHT and IPFS," *KSII Transactions on Internet & Information Systems*, vol. 16, no. 7, 2022.

[84] L. Balduf, S. Henningsen, M. Florian, S. Rust, and B. Scheuermann, "Monitoring data requests in decentralized data storage systems: A case study of IPFS," in *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*, 2022: IEEE, pp. 658-668.

[85] H. Huang, J. Lin, B. Zheng, Z. Zheng, and J. Bian, "When blockchain meets distributed file systems: An overview, challenges, and open issues," *IEEE Access*, vol. 8, pp. 50574-50586, 2020.

[86] R. Kumar, R. Tripathi, N. Marchang, G. Srivastava, T. R. Gadekallu, and N. N. Xiong, "A secured distributed detection system based on IPFS and blockchain for industrial image and video data security," *Journal of Parallel and Distributed Computing*, vol. 152, pp. 128-143, 2021.

[87] G. S. Reen, M. Mohandas, and S. Venkatesan, "Decentralized patient centric e-Health record management system using blockchain and IPFS," in *2019 IEEE Conference on Information and Communication Technology*, 2019: IEEE, pp. 1-7.

[88] R. Kumar and R. Tripathi, "Implementation of distributed file storage and access framework using IPFS and blockchain," in *2019 Fifth International Conference on Image Information Processing (ICIIP)*, 2019: IEEE, pp. 246-251.

[89] P. Kang, W. Yang, and J. Zheng, "Blockchain Private File Storage-Sharing Method Based on IPFS," *Sensors*, vol. 22, no. 14, p. 5100, 2022.

[90] S. Samonas and D. Coss, "The CIA strikes back: Redefining confidentiality, integrity and availability in security," *Journal of Information System Security*, vol. 10, no. 3, 2014.
[91] E. O. Yeboah-Boateng, *Cyber-security challenges with smes in developing economies: Issues of confidentiality, integrity & availability (CIA)*. Institut for Elektroniske Systemer, Aalborg Universitet, 2013.
[92] M. Warkentin and C. Orgeron, "Using the security triad to assess blockchain technology in public sector applications," *International Journal of Information Management*, vol. 52, p. 102090, 2020.
[93] A. Maetouq, S. M. Daud, N. A. Ahmad, N. Maarop, N. N. A. Sjarif, and H. Abas, "Comparison of hash function algorithms against attacks: A review," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 8, 2018.
[94] E. A. Adeniyi, P. B. Falola, M. S. Maashi, M. Aljebreen, and S. Bharany, "Secure sensitive data sharing using RSA and ElGamal cryptographic algorithms with hash functions," *Information*, vol. 13, no. 10, p. 442, 2022.
[95] B. Halak, Y. Yilmaz, and D. Shiu, "Comparative analysis of energy costs of asymmetric vs symmetric encryption-based security applications," *IEEE Access*, vol. 10, pp. 76707-76719, 2022.
[96] R. Anusha, M. D. Kumar, V. S. Shetty, and N. P. Hegde, "Symmetric Key Algorithm in Computer security: A Review," in *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2020: IEEE, pp. 765-769.
[97] J. Kapoor and D. Thakur, "Analysis of Symmetric and Asymmetric Key Algorithms," in *ICT Analysis and Applications*, 2022: Springer, pp. 133-143.
[98] R. Karim, L. S. Rumi, M. Ashiqul Islam, A. A. Kobita, T. Tabassum, and M. Sagar Hossen, "Digital signature authentication for a bank using asymmetric key cryptography algorithm and token based encryption," in *Evolutionary Computing and Mobile Sustainable Networks: Proceedings of ICECMSN 2020*, 2021: Springer, pp. 853-859.
[99] L. Abualigah, M. Abd Elaziz, P. Sumari, Z. W. Geem, and A. H. Gandomi, "Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer," *Expert Systems with Applications*, vol. 191, p. 116158, 2022.
[100] A. Hamza and B. Kumar, "A review paper on DES, AES, RSA encryption standards," in *2020 9th International Conference System Modeling and Advancement in Research Trends (SMART)*, 2020: IEEE, pp. 333-338.

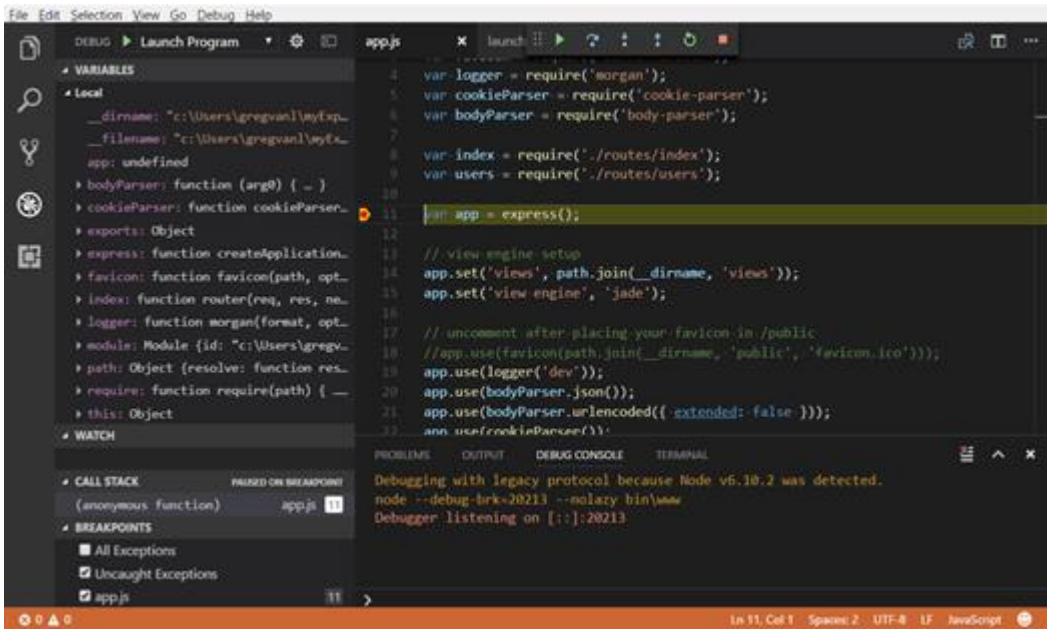
[101] X. Zhou and X. Tang, "Research and implementation of RSA algorithm for encryption and decryption," in *Proceedings of 2011 6th international forum on strategic technology*, 2011, vol. 2: IEEE, pp. 1118-1121.
[102] F. J. Aufa and A. Affandi, "Security system analysis in combination method: RSA encryption and digital signature algorithm," in *2018 4th International Conference on Science and Technology (ICST)*, 2018: IEEE, pp. 1-5.
[103] T. M. Zaw, M. Thant, and S. Bezzateev, "Database security with AES encryption, elliptic curve encryption and signature," in *2019 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF)*, 2019: IEEE, pp. 1-6.
[104] Y. Genç and E. Afacan, "Implementation of new message encryption using elliptic curve cryptography over finite fields," in *2021 International Congress of Advanced Technology and Engineering (ICOTEN)*, 2021: IEEE, pp. 1-6.
[105] N. J. G. Saho and E. C. Ezin, "Comparative study on the performance of elliptic curve cryptography algorithms with cryptography through RSA algorithm," in *CARI 2020-Colloque Africain sur la Recherche en Informatique et en Mathématiques Apliquées*, 2020.
[106] M. Poongodi *et al.*, "Prediction of the price of Ethereum blockchain cryptocurrency in an industrial finance system," *Computers & Electrical Engineering*, vol. 81, p. 106527, 2020.
[107] S. K. Panda and S. C. Satapathy, "An investigation into smart contract deployment on Ethereum platform using Web3. js and solidity using blockchain," in *Data Engineering and Intelligent Computing: Proceedings of ICICC 2020*, 2021: Springer, pp. 549-561.
[108] G. D. H. Niranga and V. S. Nair, "Design of a Secured Medical Data Access Management Using Ethereum Smart Contracts, Truffle Suite and Web3," in *Proceedings of the Twentieth ACM Conference on Embedded Networked Sensor Systems*, 2022, pp. 1215-1221.
[109] L. Cao, "Decentralized ai: Edge intelligence and smart blockchain, metaverse, web3, and desci," *IEEE Intelligent Systems*, vol. 37, no. 3, pp. 6-19, 2022.
[110] D. Pramulia and B. Anggorojati, "Implementation and evaluation of blockchain based e-voting system with Ethereum and Metamask," in *2020 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, 2020: IEEE, pp. 18-23.

[111] K. Bhosale, K. Akbarabbas, J. Deepak, and A. Sankhe, "Blockchain based secure data storage," *International Research Journal of Engineering and Technology (IRJET),* vol. 6, no. 3, pp. 5058-5061, 2019.

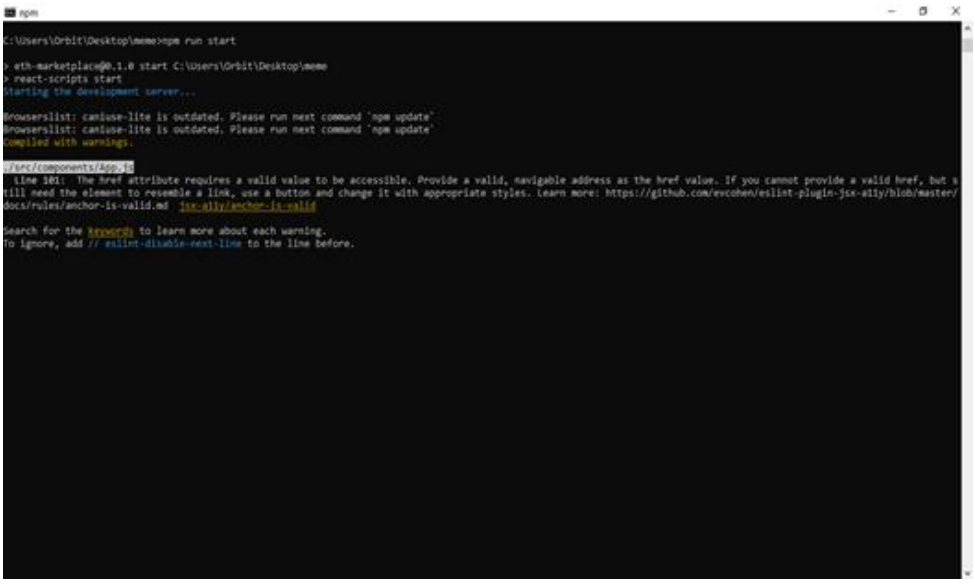
Appendix

1- A- the interface of writing to node.js while writing the instructions inside it. Where it is written by CMD.

B- *npm* is a command for JavaScript; it allows you to install any library inside JavaScript,

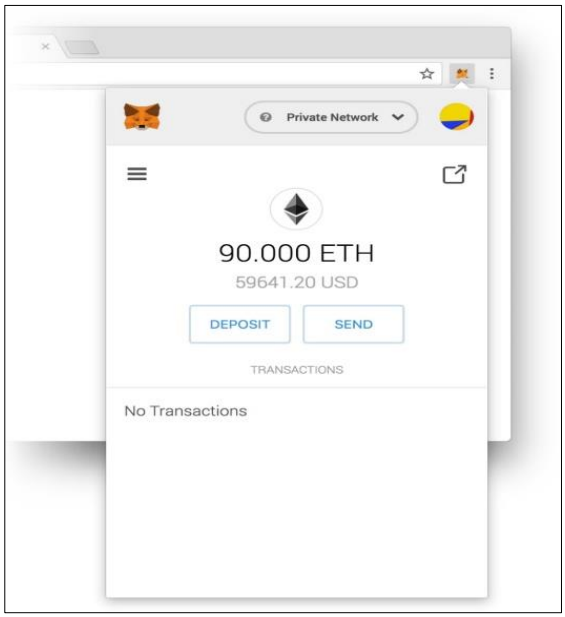


A) Website of Node.js



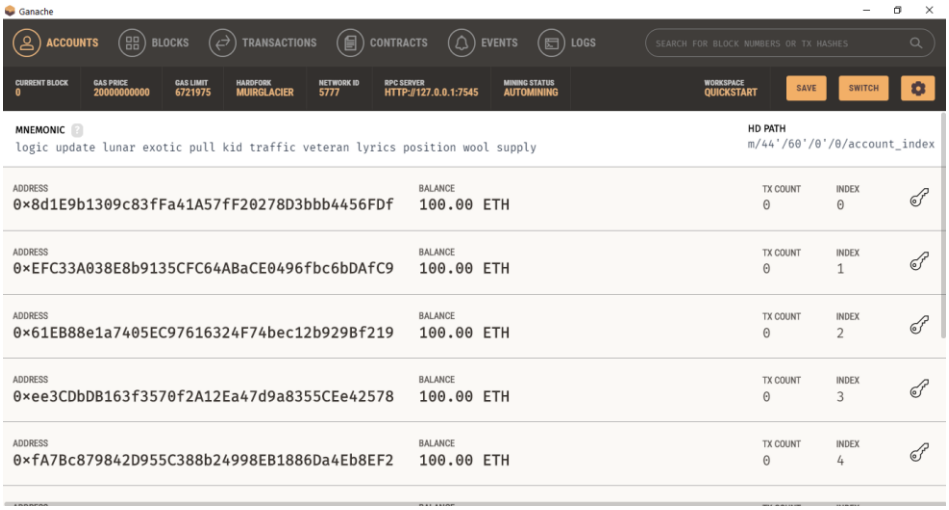
B) Window of npm in Node.js

2- Meta mask(wallet in blockchain)account of patent



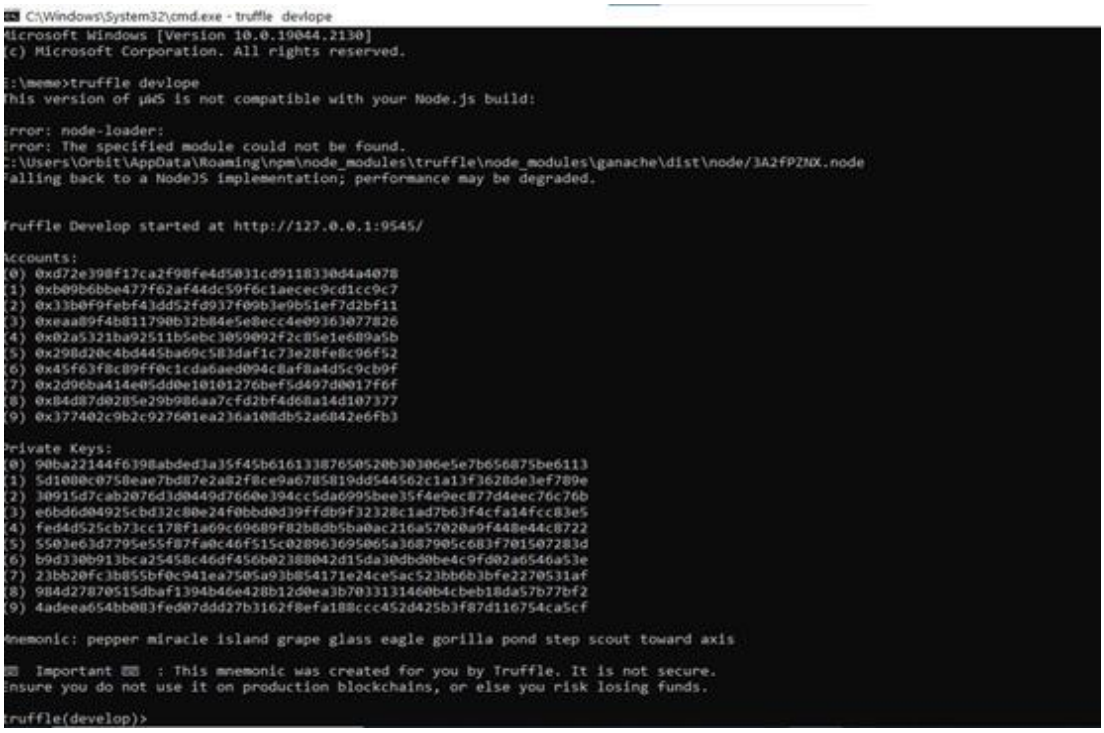
MetaMask in the browser.

3- Ganache Ethereum test network to deploy smart

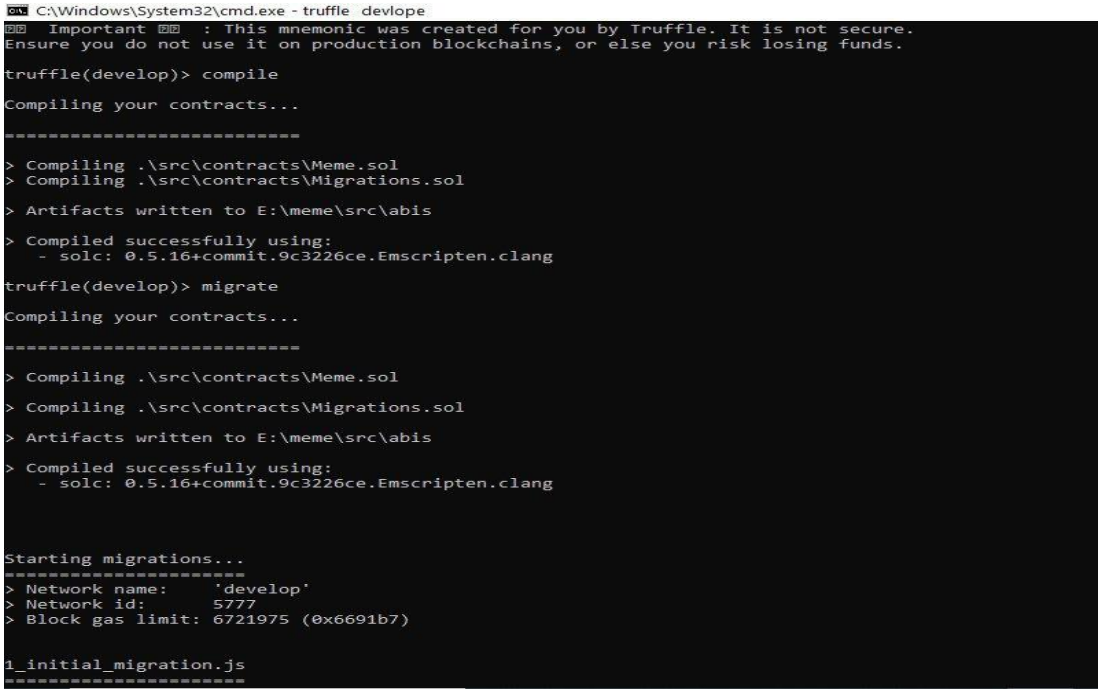


Ganache platform

4- Implementation steps of storing process



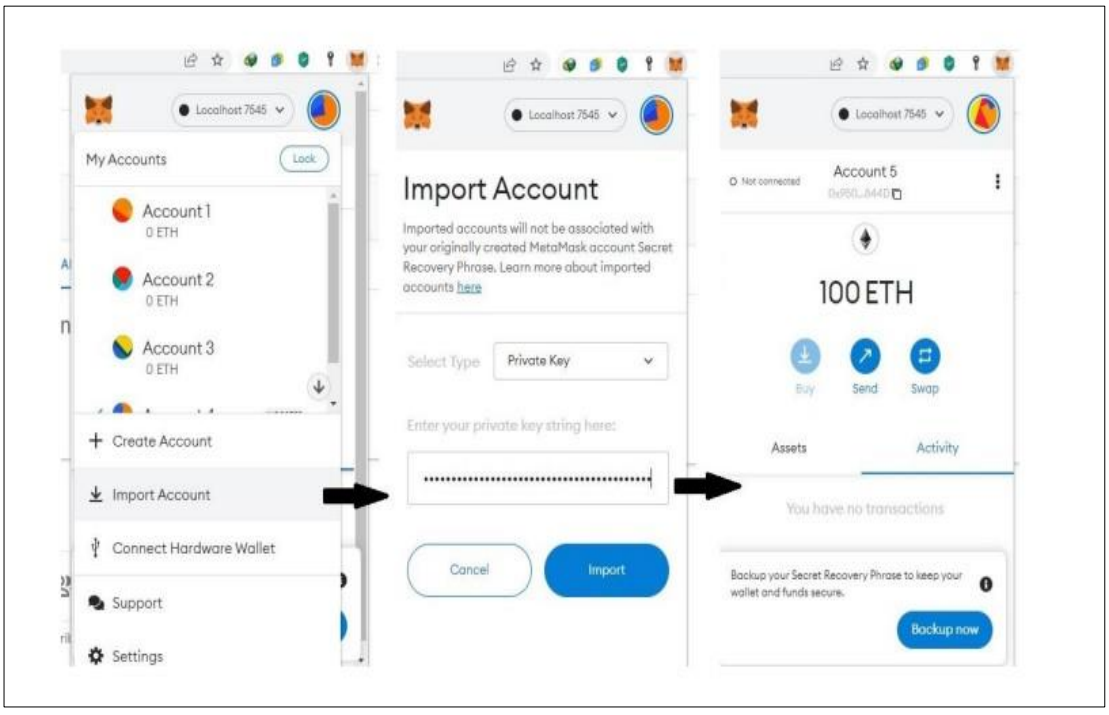
Step1 : Truffle development



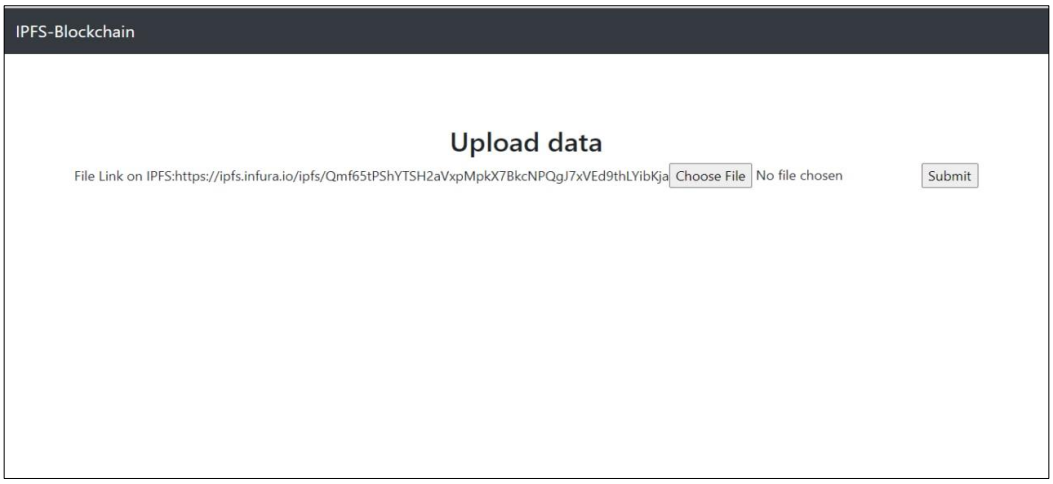
Step 2: Compile the smart contract.

```
C:\Windows\System32\cmd.exe - truffle  devlope
   > value sent:           0 ETH
   > total cost:           0.000764562375 ETH


   > Saving migration to chain.

   > Saving artifacts
   -------------------------------------
   > Total cost:      0.000764562375 ETH


2_deploy_contract.js
====================


   Replacing 'Meme'
   ----------------

   > transaction hash:    0xa70359f41e31f462c89d791785ef2d8f6459760a385710fdd386a45f54c45cea

   > Blocks: 0            Seconds: 0
   > contract address:    0x339a578d6116882de8bb98876a8096301A2f78E1
   > block number:        3
   > block timestamp:     1679692747
   > account:             0xd72E398f17Ca2f98fE4d5031CD9118330D4A4078
   > balance:             99.998379401571487538
   > gas used:            222254 (0x3642e)
   > gas price:           3.177688086 gwei
   > value sent:          0 ETH
   > total cost:          0.000706253887865844 ETH


   > Saving migration to chain.

   > Saving artifacts
   -------------------------------------
   > Total cost:     0.000706253887865844 ETH

Summary
=======
> Total deployments:   2
> Final cost:          0.001470816262865844 ETH
```

Step 4: Migration of the smart contract.

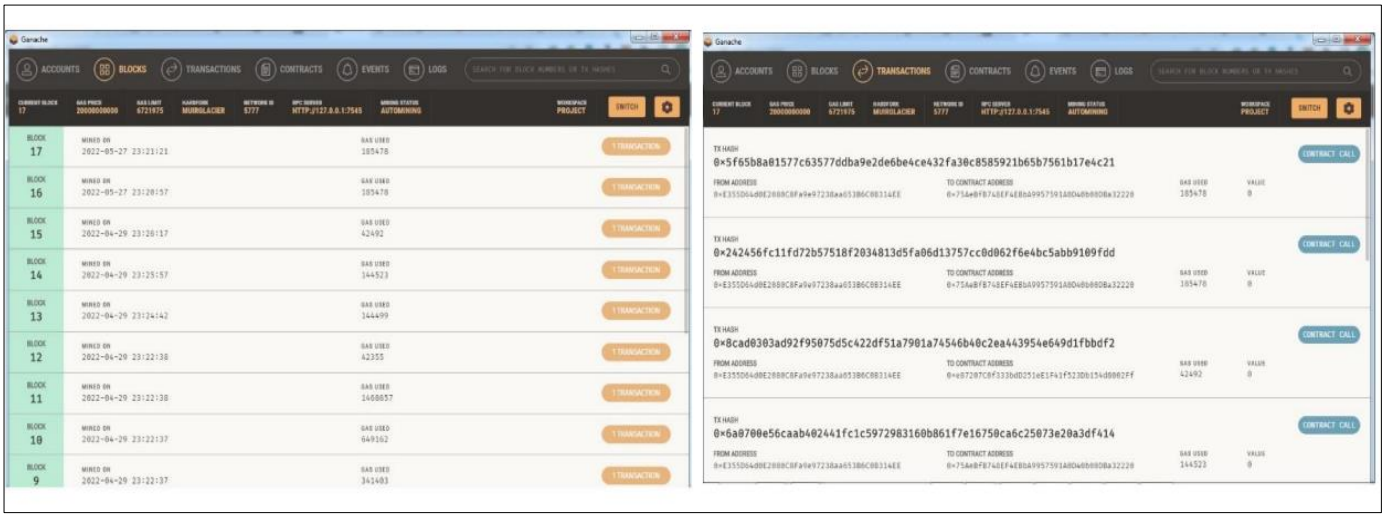


Step 5: Steps for creating a wallet.



Step 6: Upload files to hospital website.

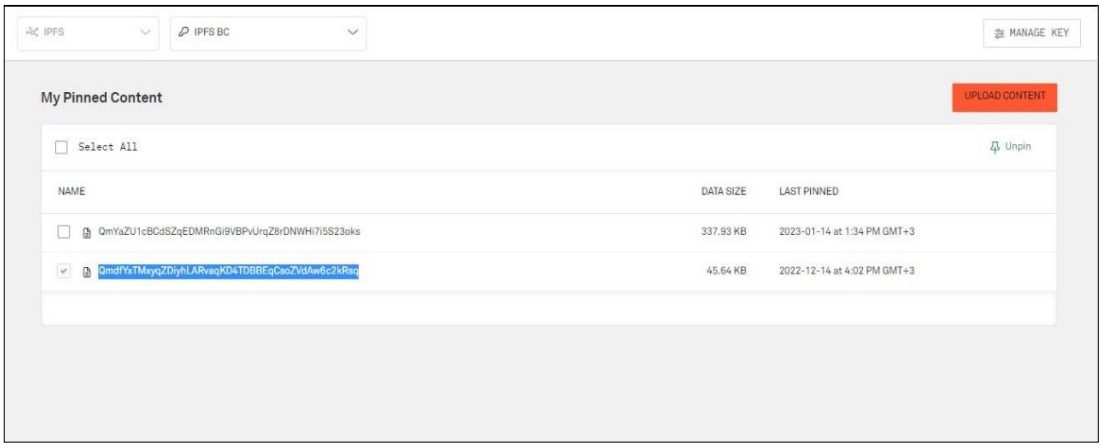
Step 7: shows the page of blocks and transactions within the server Ganache



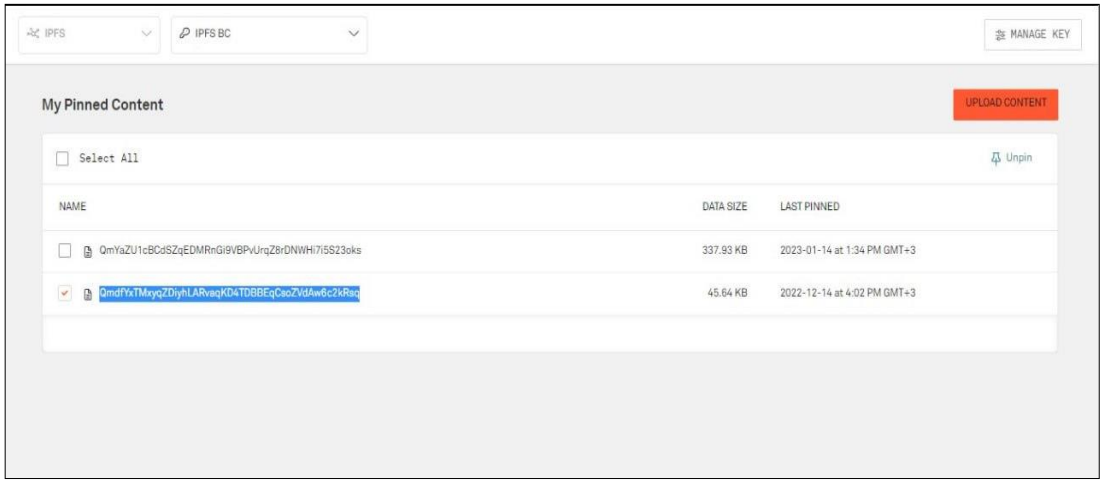
(A) Books (B)Transaction

.

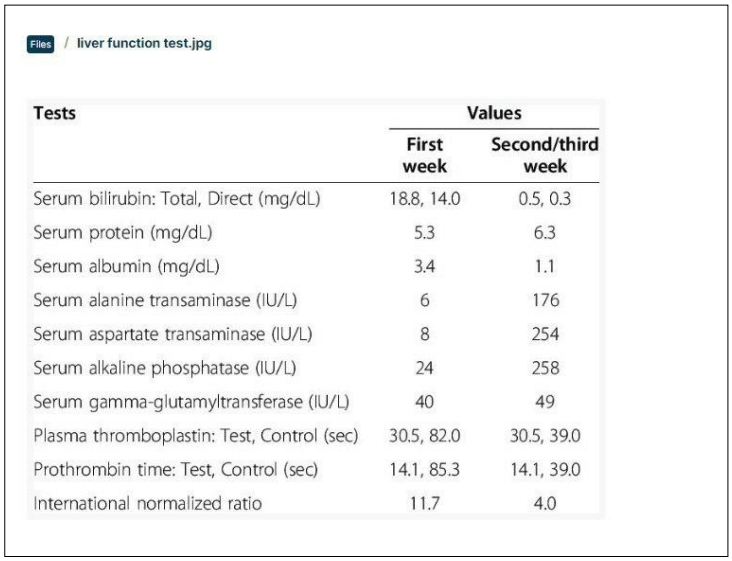
5- implementation steps of retrieving process



Step1: select the CID file



Step 2: Retrieving files


Files / liver function test.jpg

Tests	Values	
	First week	Second/third week
Serum bilirubin: Total, Direct (mg/dL)	18.8, 14.0	0.5, 0.3
Serum protein (mg/dL)	5.3	6.3
Serum albumin (mg/dL)	3.4	1.1
Serum alanine transaminase (IU/L)	6	176
Serum aspartate transaminase (IU/L)	8	254
Serum alkaline phosphatase (IU/L)	24	258
Serum gamma-glutamyltransferase (IU/L)	40	49
Plasma thromboplastin: Test, Control (sec)	30.5, 82.0	30.5, 39.0
Prothrombin time: Test, Control (sec)	14.1, 85.3	14.1, 39.0
International normalized ratio	11.7	4.0

Step 3: Preview files after retrieving prosses.

الخلاصة

يشكل الحفاظ على سرية الملفات الصحية تحديًا كبيرًا بسبب تضمين بيانات حساسة .تنشأ مخاوف بشأن الحفاظ على الملفات الورقية ، وحماية الخصوصية أثناء عملية الرقمنة ، واحتمال الوصول والتلاعب غير المصرح به .تقدم هذه الأطروحة اقتراحًا لمعالجة هذه المخاوف من خلال استخدام تقنيات واستراتيجيات ومناهج التشفير والبلوك تشين و .IPFS تم اختبار الحل المقترح وتحليله بدقة ، والتأكد من صحته .يستفيد النهج المستخدم من العقود الذكية والثبات المتأصل في تقنية blockchain لتعزيز أمان وسلامة عمليات استرجاع البيانات .تتيح شبكة IPFSللمستخدمين مشاركة البيانات بأمان .تقدم هذه الأطروحة المنهجيات التي تستخدم blockchainو IPFS (نظام الملفات بين الكواكب) لتحقيق إخفاء هوية البيانات الحساسة .علاوة على ذلك ، قدم أدلة تجريبية توضح فعالية هذه الأساليب وعمليتها من خلال سلسلة من التجارب . يوفر نظام تخزين بيانات الرعاية الصحية اللامركزي الذي قمنا بتطبيقه العديد من المزايا ، بما في ذلك حماية الخصوصية المحسنة .تعالج العقود الذكية ونظام الملفات بين الكواكب (IPFS) التحديات المرتبطة باللامركزية .تُظهر الأنظمة أداءً فائقًا مقارنة بالنُهج البديلة من حيث سعة التخزين .يسمح ذلك بإنشاء شبكة blockchain خاصة ومرخصة من نظير إلى نظير تتضمن أصحاب المصلحة المسجلين والمحددين .نتيجة لذلك ، تضمن هذه الشبكة المستويات المثلى من التشغيل البيني والأمان وقابلية التوسع والترخيص .يوضح نظام الملفات بين الكواكب (IPFS) مستوى عالٍ من الكفاءة في نقل بيانات الطرف الثالث إلى خوادم البيانات .يعالج النظام بشكل فعال مجموعة واسعة من تهديدات الخصوصية مع مراعاة القيود التي تفرضها موارد blockchain.أدى ذلك إلى تطوير نظام قوي وسري للتحكم في الوصول إلى بيانات الرعاية الصحية ، والذي تم تحقيقه من خلال دمج مفاتيح التجزئة ، و IPFS ، وتكنولوجيا blockchain ، وخوارزميات التشفير .نتيجة لذلك ، وجد أن وقت تنفيذ إرسال ملف المريض بدون إطار العمل المقترح سيستغرق 9.6 ثانية .من ناحية أخرى ، فإن وقت التنفيذ باستخدام إطار العمل المقترح ، والذي استغرق 2.6 ثانية ، بالإضافة إلى ذلك ، يقلل النظام من تكلفة تخزين الملفات التي تحتاج 0.001. ETH لكل 1KB

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة بابل / كلية تكنولوجيا المعلومات
قسم شبكات المعلومات

تحسين امنية الملفات الطبية بالاعتماد على نظام الملفات بين العقد وتقنية سلسلة الكتل

رسالة مقدمة

إلى مجلس كلية تكنولوجيا المعلومات في جامعة بابل كجزء من متطلبات

الحصول على درجة الماجستير في تكنولوجيا المعلومات / شبكات المعلومات

من قبل

رنا عباس رضا رجب

بأشراف

أ.م.د الحارث عبد الكريم عبدالله

1444 هـ 2023 م