IPFS BASED FILE STORAGE ACCESS CONTROL AND AUTHENTICATION MODEL FOR SECURE DATA TRANSFER USING BLOCKCHAIN TECHNIQUE

Project Submitted to the
SRM University AP, Andhra Pradesh
for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology
in
Computer Science & Engineering
School of Engineering & Sciences

submitted by

Gowtham Krishna Yarra(AP20110010007)

Srinivasarao Botla(AP20110010014)
Nikhil Chowdary Yamani(AP20110010067)

Under the Guidance of

Prof. Deepthi K C Kakumani

Department of Computer Science & Engineering
SRM University AP
Neerukonda, Mangalgiri, Guntur
Andhra Pradesh - 522 240
May 2024

DECLARATION

I undersigned hereby declare that the project report **IPFS Based File Storage Access Control and Authentication Model for Secure Data Transfer using Blockchain Technique** presented for partial completion of Bachelor of Technology degree requirements at Computer Science & Engineering, SRM University AP, is a real work done by the author under supervision of Prof. Deepthi K C Kakumani. The report expresses the author's thoughts in their own words, with proper citations and references for any additions. The author upholds academic honesty and integrity by not misrepresenting or fabricating any data, concept, fact, or source.Violations may result in disciplinary proceedings from the educational institution or university, in addition to legal action if sources are not properly cited or permission is not obtained. The study has not used as the basis for any future university degree awarding.

Place	:	Date	: May 12, 2024
Name of student	: Gowtham Krishna Yarra	Signature	: .....
Name of student	: Srinivasarao Botla	Signature	: .....
Name of student	: Nikhil Chowdary Yamani	Signature	: .....

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SRM University AP

Neerukonda, Mangalgiri, Guntur

Andhra Pradesh - 522 240

CERTIFICATE

This is to certify that the report entitled **IPFS Based File Storage Access Control and Authentication Model for Secure Data Transfer using Blockchain Technique** submitted by to the SRM University AP in partial fulfillment of the requirements for the award of the Degree of Master of Technology in in is a bonafide record of the project work carried out under my/our guidance and supervision. This report, in any form, has not been transmitted to another university or institute for whatever reason.

Project Guide	Head of Department
Name : Prof. Deepthi K C Kakumani	Name : Prof. Niraj Upadhayaya
Signature:	Signature: .....

ACKNOWLEDGMENT

I am grateful to everyone who assisted me prepare and deliver this Project Report named **IPFS Based File Storage Access Control and Authentication Model for Secure Data Transfer using Blockchain Technique**.

I am grateful to my guide and supervisor Prof. Deepthi K C Kakumani at the Department of Computer Science & Engineering for their useful ideas and critical input in the production of this report. I am grateful to Prof. Niraj Upadhayaya, Head of Department of Computer Science & Engineering for his encouragement.

My classmates have been supportive and have patiently listened to my project-related presentations.

Gowtham Krishna Yarra (AP20110010007)
Srinivasarao Botla(AP20110010014)
Nikhilchowdary Yamani (AP20110010067)
B. Tech.
Department of Computer Science & Engineering
SRM University AP

ABSTRACT

Centralized servers provide serious threats to the security and privacy of data in the current paradigm of online data exchange. Secure, decentralized data storage with fine-grained access control is proposed in this study using Blockchain technology and IPFS (Interplanetary File System). The solution guarantees that data stays secure, unchangeable, and accessible only to authorized users by merging Attribute-Based Encryption (ABE) with Blockchain. Data Owners upload and share files on IPFS. Data Requesters request access to these files. Solidity smart contracts handle application operations. Blockchain is used for distributed data management. These five components form the system. Included in the eight-step procedure are the following: setup, registration, initialization, authentication, testing, data encryption and storage, and access control. This concept eliminates the need for users to depend on insecure central servers for sharing and accessing data. The suggested solution creates a strong foundation for safe and decentralized data sharing by merging IPFS, ABE, and Blockchain. This helps to reduce the hazards of centralized storage and allows for fine-grained access control. The suggested work expands upon previous efforts by substituting the computationally demanding Chebyshev algorithm with the faster and more secure CHACHA20 method, which increases speed and dependability by 70%.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION TO THE PROJECT

More and more people are turning to blockchain technology as a way to store databases decentralizedly [21]. In addition, blockchain technology has features such as an immutable distributed ledger that records all transactions on a specific network. Several businesses, including healthcare, supply chain management, and the financial sector, are already embracing the blockchain technology that Satoshi Nakamoto used to create bitcoin [22]. The globe over, a plethora of distributed file storage applications are now in use. The storage and access issues that arise when collaborating with different network partners are a challenge for these programs. A popular file-sharing and storage application, BitTorrent gives users the capabilities they need to interact with untrustworthy sender network partners [23]. The most noteworthy and crucial aspect of BitTorrent is the fact that it has 170 million regular users [24]. Furthermore, BitTorrent gatherings often account for 40% of all online traffic [24]. Nevertheless, Hypertext Transfer Protocol (HTTP) becomes more competent than this application. The Hypertext Transfer Protocol (HTTP) is without a doubt the most remarkable standard when considering the global reach of communication applications. Distributed file admission is not well-suited to the location-based standards upon which the HTTP protocols depend. We eliminate BitTorrent and HTTP issues and construct a better web by using IPFS, a decentralized file storage network [25]. The IPFS uses the substance-addressed approach for both file storage and reading [26].

OBJECTIVE:

The purpose of this article is to provide a decentralized and safe method for exchanging data that makes use of Blockchain and IPFS. The goal is to provide granular access control by combining Attribute-Based Encryption (ABE) with Smart Contracts; this will guarantee data privacy, immutability, and resistance against weaknesses in centralized servers.

DESCRIPTION:

1. Because of their susceptibility to hacking and illegal access, centralized servers endanger data privacy and security.
2. The current system for exchanging data does not have granular control over who may access it, which might cause data exposure or interruptions in service.
3. Data privacy and security are compromised, which affects users who share data online.
4. Exposed data, interrupted services, and distrust are the outcomes of centralized server vulnerabilities.
5. Our decentralized data sharing solution guarantees data privacy, immutability, and resilience against server vulnerabilities.The system includes Attribute-Based Encryption for specific control of access and utilizes Blockchain and IPFS.

1.1 SOFTWARE REQUIREMENTS

In order for a program to run well on a computer, certain software resources and installation requirements must be defined. In most cases,

you will need to install these prerequisites or needs independently of the program installation package.

Platform – A platform is a framework, in computing, that enables software to execute. This framework might be hardware- or software-based. Platforms often include the hardware, software, and operating system of a computer, as well as any languages and their associated runtime libraries. Among the first items included when outlining system requirements (software) is the operating system. While it is common practice to ensure software is backward compatible with older versions of operating systems, there may be instances when this is not the case. For instance, although it is not always the case, Windows 98 will not run most of the applications that have been developed for Windows XP. Similarly, Linux distributions utilizing Kernel v2.2 or v2.4 will not execute or build software built using the newest capabilities of Linux Kernel v2.6.

APIs and drivers – Software that depends significantly on specialist hardware, such as premium display adapters, requires either upgraded device drivers or specialist application programming interfaces (APIs). One such example is Microsoft DirectX, a collection of application programming interfaces (APIs) to control multimedia-related activities, primarily game creation, on Windows and other Microsoft platforms.

Web browser – The default web browser on most computers is used by the majority of online applications and software that rely significantly on Internet technology. Many people still use Microsoft Internet Explorer on Windows, even though it uses the vulnerable ActiveX components.

1) PYTHON IDLE (3.7.0)

2) Node JS

3) Visual Studio Community Version

4) LANGUAGES BACK-END: Python, Java Script, Solidity

5) LANGUAGES FRONT-END: HTML, CSS, JS, Boot Strap

1.2 HARDWARE REQUIREMENTS

Physical assets of a computer, or the equipment, are the most basic set of needs for any software program or operating system. In the framework of operating systems, the list of hardware compatibility (HCL) is usually added with a hardware specifications list. An HCL is a collection of certified, compatible and sometimes conflicting hardware components for the specified operating system or software. Different aspects of the hardware requirements are discussed in the following sections.

Machine Architecture – To work with a specific kind of machine, each operating system is designed. Almost all programs have strict requirements about the operating systems and hardware architectures that they can only operate on. It is necessary to recompile the majority of software and operating systems in order to make them work on a different architecture, even if such systems do exist. Also, have a look at this compilation of popular OSes and the architectures that power them.

Processing power – TSoftware requires a system with sufficient processing capacity, which is directly related to the central processing unit (CPU). In software running on the x86 architecture, processor performance is typically determined by CPU model and clock speeds. It is common practice to disregard the many other CPU characteristics that affect performance and power, such as bus speed, cache, and MIPS. Because even at the same clock speed, Intel Pentium and AMD Athlon CPUs may have vastly differing throughput rates, this definition of power is often incorrect. Many people think of Intel

Pentium CPUs when they hear this category, because of how popular they are.

Memory – During execution, all software is stored in computer random access memory (RAM). Memory requirements shall be determined based on the needs of the application, operating system, auxiliary files and software as well as any processes which are currently being used. The criteria shall be defined in such a way as to take into account the optimal performance of other, unrelated applications running on a multitasking system.

Secondary storage – Factors like program size, the amount of transient files created and kept during software installation and operation, as well as possible use of spare space when RAM is insufficient can be used to determine additional storage capacity for your drive.

Display adapter – Graphics editors and high-end games are examples of software that typically specify a high-end display adapter as part of the system requirements.

Peripherals – The increased performance or capability of certain peripherals is required by certain software programs, which in turn make heavy use of peripherals. Keyboards, CD-ROM drives, pointing devices, network adapters, etc. were the examples of such peripherals.

1) Operating System : Windows Only

2) Processor : i5 and above

3) RAM : 8GB and above

4) Hard Disk: 25 GB in local drive

Chapter 2

MOTIVATION

The motivation for IPFS Based File Storage Access Control And Authentication Model For Secure Data Transfer Using Blockchain Technique is as follows,

Blockchain can stem from its potential to enable decentralization, security, automation, tokenization, interoperability, community collaboration, and regulatory compliance. By harnessing the capabilities of blockchain technology, projects can address diverse use cases, drive innovation, and create value in various industries and sectors.

IPFS can revolve around its decentralized nature, data integrity features, efficient content distribution mechanisms, suitability for offline environments, integration with blockchain technology, and participation in an open and collaborative ecosystem. By leveraging the unique attributes of IPFS, projects can address various use cases and contribute to the advancement of decentralized and distributed systems.

D. Bernstein designed the ChaCha ciphers in 2008. In 2016, Google used the ChaCha cipher paired with a Message Authentication Code, poly-1305, for their TLS security protocol. Adopting this combination has increased traffic security on Chrome browsers for mobile devices and Google websites. Mozilla, a web browser, has also embraced this combination in its network security protocol. As a result, this pushes me to analyze ChaCha.[20]

Chapter 3

LITERATURE SURVEY

The literature survey highlights several innovative approaches leveraging blockchain technology to address security and integrity concerns across various domains. Multi-party authorization (MPA) systems, commonly centralized, face challenges in ensuring secure and auditable access control. A proposed decentralized blockchain-based solution employs Ethereum smart contracts for MPA and oracles for secure data access using proxy re-encryption algorithms, enhancing security and auditability. Incorporating reputation mechanisms rates oracle reliability, strengthening the system's trustworthiness.

In the finance sector, fraud prevention and secure record-keeping are paramount. A blockchain-based scheme integrates ciphertext-policy attribute-based encryption and fog nodes for efficient staff operations. Encrypted insurance records stored on the Inter Planetary File System (IPFS) ensure security and decentralization, while blockchain immutability ensures non-repudiation. The scheme's security proof demonstrates effectiveness against selected keyword attacks, with performance analyses validating its efficiency and feasibility.

Electronic medical records (EMRs) face integrity and security challenges, addressed through attribute-based encryption on IPFS combined with blockchain technology. Secure storage and efficient sharing are achieved while preserving access control and data retrieval efficiency. Blockchain's tamper-proof nature ensures secure storage and search for medical data,

with performance analyses confirming the scheme's efficiency and feasibility.

In research communities, trust issues plague existing data sharing platforms reliant on trusted third parties. A proposed blockchain-based platform, leveraging IPFS, ensures transparency, security, and immutability. Decentralized storage, smart contracts for access control, and an incentive mechanism for authentic reviews enhance trust and authenticity. Simulation results provide insights into gas consumption, cost estimates, and encryption scheme performance, validating the solution's efficacy and practicality.

In summary, the literature survey highlights the diverse applications and potential benefits of combining blockchain and IPFS. From securing access to sensitive data and ensuring data integrity to enabling efficient data sharing in decentralized environments, the integration of blockchain and IPFS presents a promising avenue for addressing the evolving challenges of data storage and management in today's digital landscape.

Chapter 4

DESIGN AND METHODOLOGY

4.1 METHODOLOGY

4.1.1 EXISTING SYSTEM:

Problems with message reliability and identity authentication during data transmission among automobiles are now plaguing vehicular ad hoc networks (VANETs). There are security vulnerabilities, including as data leakage and manipulation, with the current system that cars use to transmit sensor data to a trusted center. One possible answer to these problems is to set up a Consortium Blockchain-based Data Security Sharing and Storage System (DSSCB). Data integrity and reliability during transmission are guaranteed by this method via the use of digital signatures based on bilinear pairing for elliptic curves. At its core, DSSCB is based on consortium blockchain technology, which provides a secure, decentralized database that is updated by network nodes. During data transmission and storage, smart contracts are used to regulate triggering circumstances for specified nodes and to distribute data coins to participating cars. Data storage and sharing is made more secure with DSSCB, according to security and performance tests. With DSSCB, the confirmation time for data blocks is much reduced, and transmission efficiency is much improved, as compared to conventional blockchain systems.

The current system has a number of drawbacks, including the follow-

ing:

• The centralized storage technique makes it easy for manipulation and data leakage to occur.

• Data block confirmation times might be longer if we stick with older blockchain technologies. Without smart contracts, we have less say over what circumstances trigger data transfer and storage.

• There is a risk to data security and integrity due to a lack of granular access control.

4.1.2 SUGGESTED FRAMEWORK:

Using Blockchain and IPFS, the suggested solution establishes a decentralized data exchange framework that is both safe and efficient. The Interplanetary File System (IPFS) is a decentralized system that allows users to upload and exchange encrypted data. When users request a file, IPFS delivers a unique hash code. Data owners may create encryption keys using shared user credentials and The use of Attribute-Based Encryption (ABE) allows for precise control of access. This way, only authorized users will be able to decode and access the files. Programmatic tasks like user authentication, permission management, and the storing of file hash codes on the Blockchain are handled via Solidity-implemented smart contracts. Setup, registration, initialization, authentication, testing, data encryption and storage, and access control are the eight steps that the system goes through. In order to make online data sharing settings more secure, private, and resistant to hacking, the suggested solution combines IPFS, ABE, and Blockchain. This will help reduce the risks associated with centralized servers. The suggested work expands upon previous efforts by substituting the computationally demanding Chebyshev algorithm with the faster and

more secure CHACHA20 method, which increases speed and dependability by 70%.

4.1.2.(i) System Proposed Benefits:

Reduces exposure to data breaches by making use of IPFS's decentralized storage. Smart contracts allow for precise control over data transmission and access conditions. Integration of Blockchain and IPFS technologies enhances data privacy, immutability, and resilience against unauthorized access. Attribute-Based Encryption is used for fine-grained control over access, which improves data security.

4.1.3 STATUTORY NEEDS

1. User

2. Securely Shared Information

3. see messages that have been shared.

4. Graph for Storage Overhead

4.2 DESIGN

4.2.1 SYSTEM ARCHITECTURE:

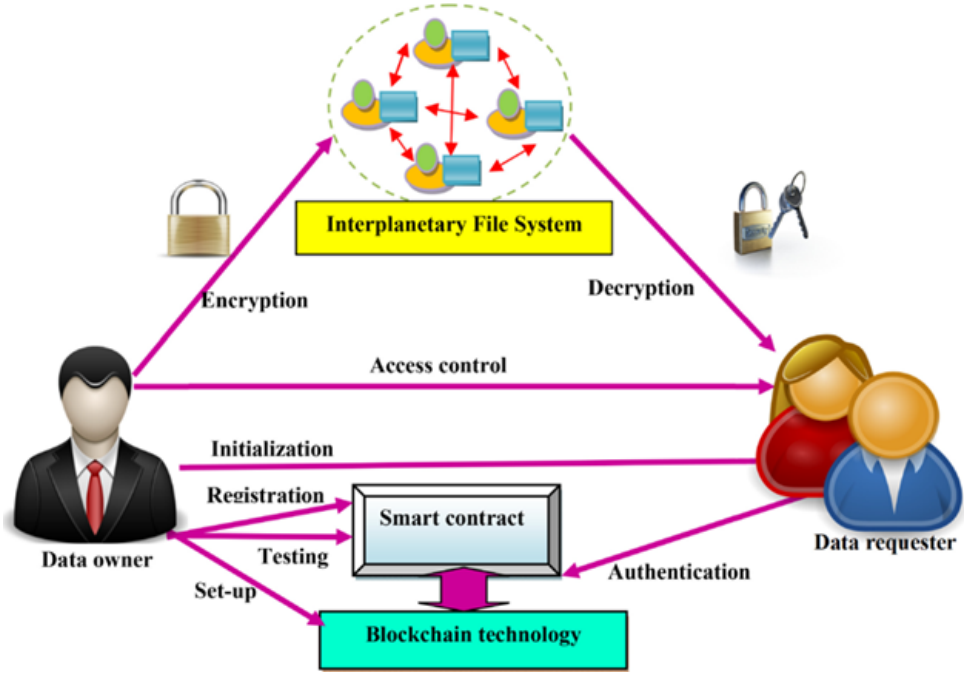


Figure 4.1: System architecture.

DATA FLOW DIAGRAM:

1. Another name of Data Flow Diagram is the bubble chart. It is a straightforward visual formalism for depicting the system consists of the data entered in it, the operations carried out with that data and the data which is released from this system.

2. Among the most crucial modeling tools of data flow diagram (DFD). It is used to model components of the system. System processes, data used by processes, external entities interacting with systems, and information flows inside systems make up these components.

3. The DFD will reveal the journey of information through the system, and how it undergoes changes. This graphical method shows the changes that happen to data as it goes from input to output and how the information

flows through the system.

4. A DFD can represent any level of abstraction.DFD may be classified into tiers based on the quantity of data movement and operational intricacy.

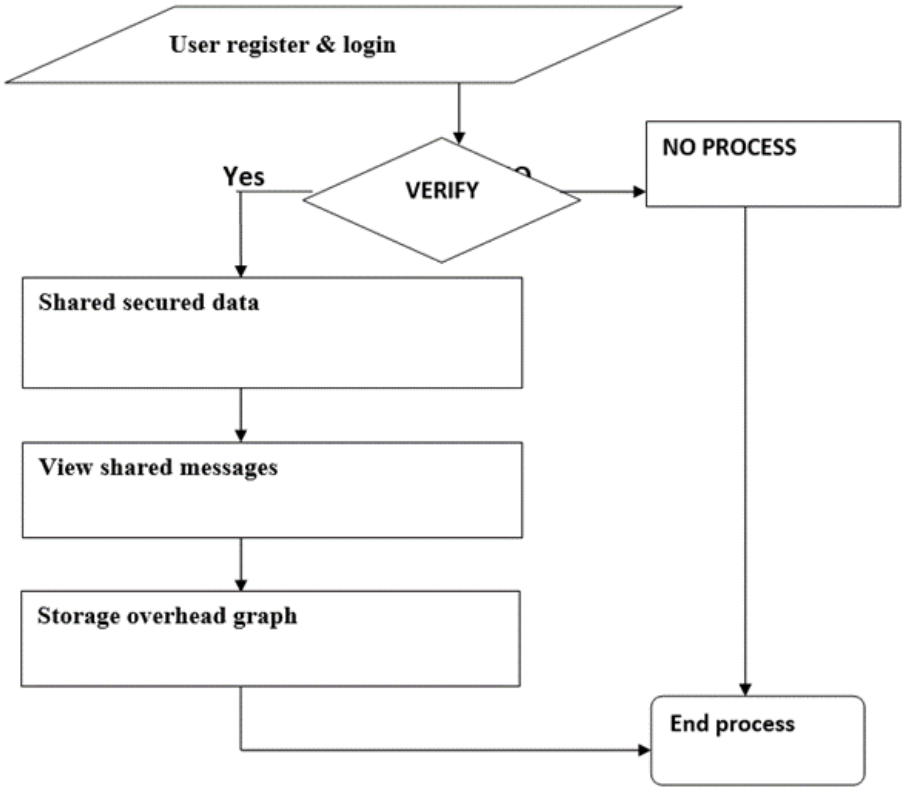


Figure 4.2: Dataflow diagrams.

4.2.2 UML DIAGRAMS

The acronym UML stands for "Unified Modeling Language". Object-oriented software developers rely on UML, a standardized multipurpose programming language. The standard is overseen by the Object Management Group, which was also responsible for its design.

The ultimate aim is to replace other languages used to represent Object Oriented Software with UML. Right now, the two main parts of UML are a meta model and a notation. Some kind of procedure or technique may likewise

be linked to or added to UML at a later date.Any system, not only software, may have its artifacts specified, visualized, constructed, and documented using the Unified Modeling Language. This includes business models as well.

The Unified Modeling Language, more commonly known as UML, represents optimal engineering methods for modeling large, complex systems.The Unified Modeling Language (UML) is heavily used in both development of software and object-oriented development of software. The UML is a software project design language that primarily makes use of visual notations.

PURPOSE:

The following were the main objectives while developing the UML:

1.Allow people to create and share meaningful models with an expressive, easy-to-use visual modeling language.

2. Give ways to make the key notions more flexible and to specialize them.

3. Not be tied down to any one development environment or programming language.

4. Give a formally sound foundation for learning the language of modeling.

5. Motivate the market for Object Oriented tools to expand.

6. Provide backing for more advanced development ideas including components, frameworks, and partnerships.

7. Combine the most effective methods.

USE CASE DIAGRAM:

A type of behavioural diagram known as a use case diagram can be defined and created by UML, or Unified Modeling Language. It tries to provide a visual picture of the system's capabilities in relation to its players, their objectives as represented by use cases, and any linkages between them.

Identifying the actors responsible for carrying out certain system operations is a key objective of use case diagrams. The system's actors may have their roles portrayed.

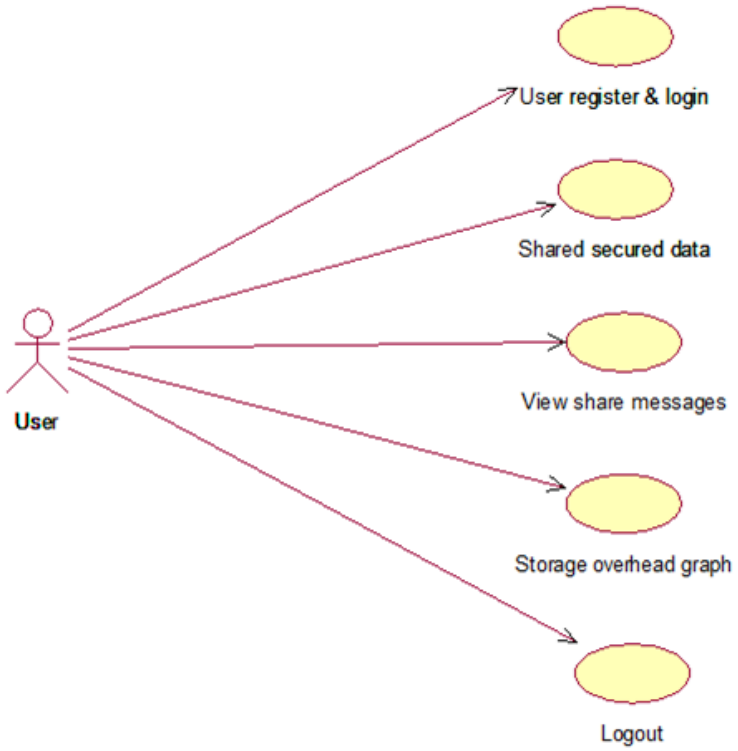


Figure 4.3: Use case diagram.

4.2.3 CLASS DIAGRAM

The class diagram enhances the use case diagram by describing the system's architecture in detail. The use case diagram represents actors, whereas the class diagram arranges them into interrelated classes. Both ”is-a” and ”has-a” relationships may describe the nature of the connection between the classes. The classes shown in the class diagram could each have the ability to do certain tasks. The ”methods” of a class are the tools it uses to accomplish its goals. In addition to this, there may be a set of ”attributes” that are exclusive to a certain class.

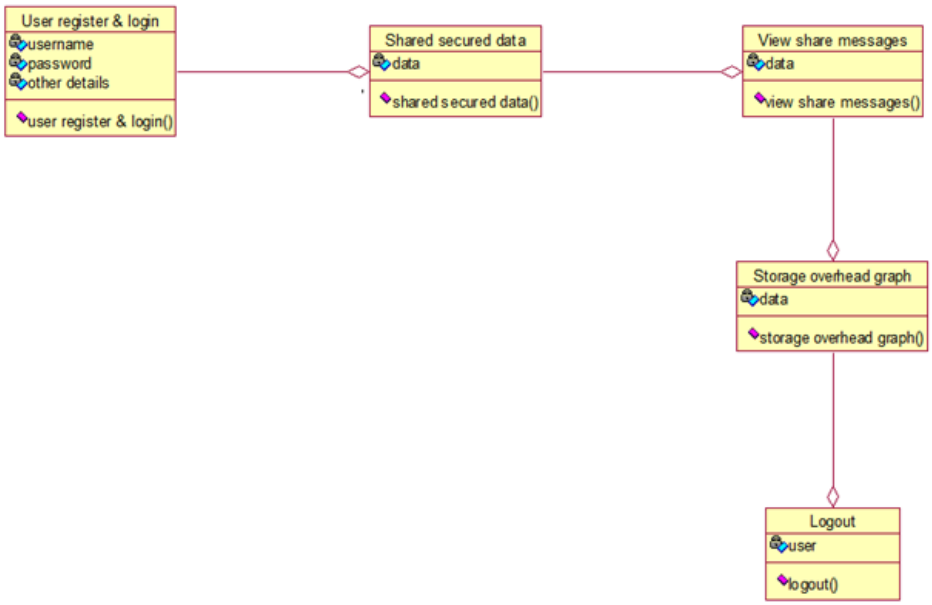


Figure 4.4: Class diagram.

4.2.4 ACTIVITY DIAGRAM:

The activity diagram shows the information about how the system's processes operate. Comparable to The activity diagram demonstrates the way the system's flow of processes. An activity diagram, like a state diagram, comprises activities, actions, transitions, initial and ending states, and security conditions.

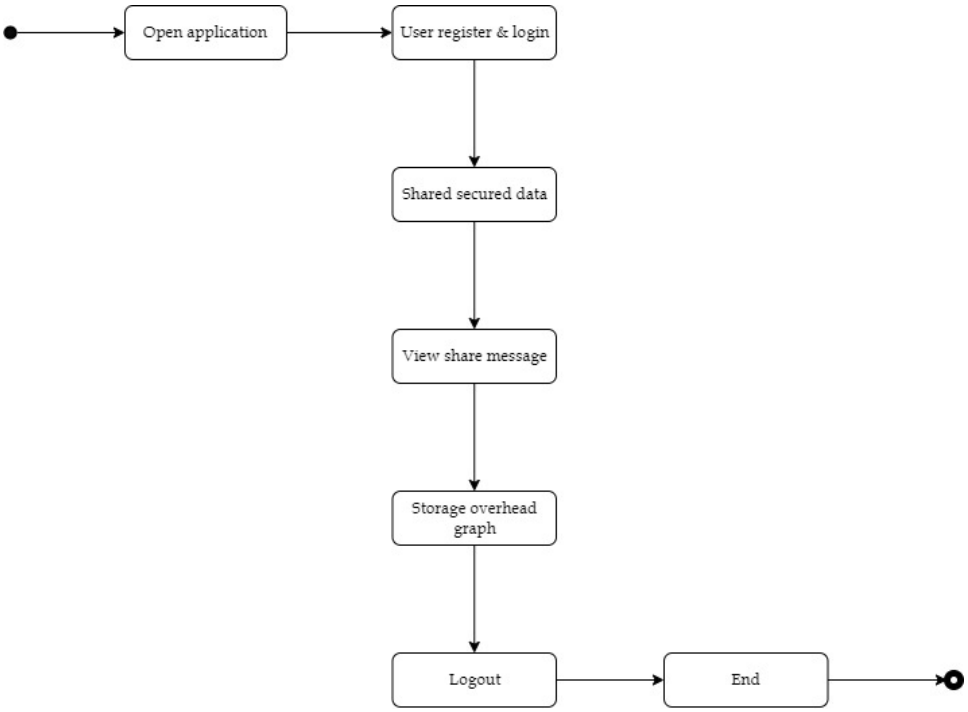


Figure 4.5: Activity diagram.

4.2.5 SEQUENCE DIAGRAM:

A sequence diagram depicts how things interact inside a system. Sequence diagrams are time-ordered. The interactions between things are illustrated step by step. Components in the sequence diagram connect through ”messages”.

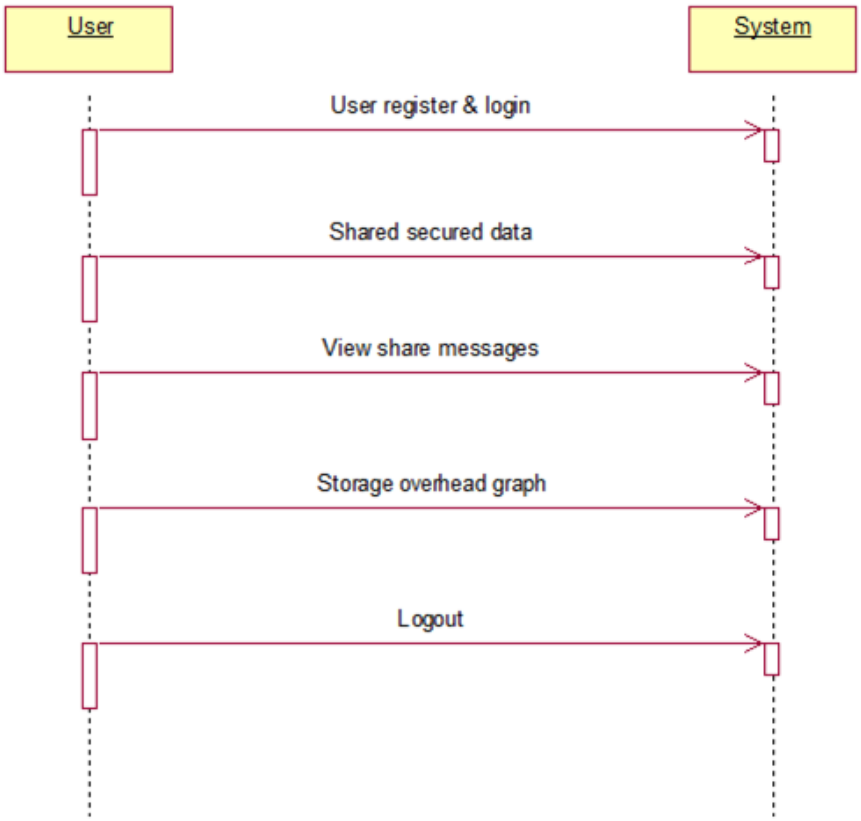


Figure 4.6: Sequence diagram.

4.2.6 COLLABORATION DIAGRAM

A cooperation diagram graphically depicts the relationships between many entities. To facilitate tracing their order, the interactions are presented as numbered interactions. Each object's potential interactions with other things may be better understood with the use of the collaboration diagram.

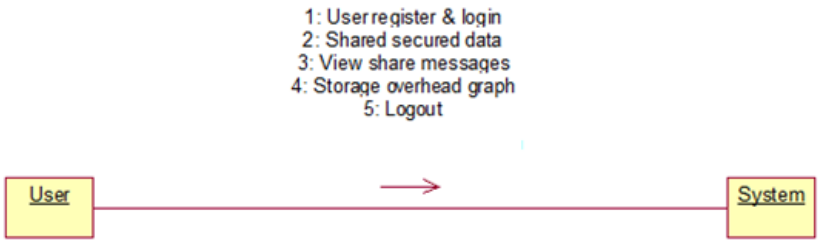


Figure 4.7: Collaboration diagram.

4.2.7 COMPONENT DIAGRAM

You can see the big picture of the system's components in the component diagram. This schematic shows the major parts of the system and their connections at a high level. Once the development or building phase of a system is complete, the components that were removed may be shown in a component diagram.

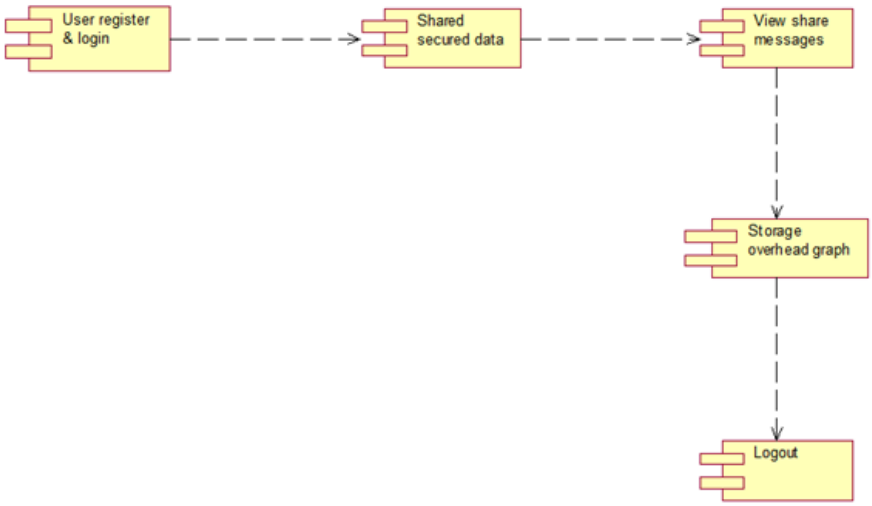


Figure 4.8: Component diagram.

Chapter 5

IMPLEMENTATION

In the event of a server attack, all user data would be exposed, and services would be disrupted, since we often communicate data online and all of this data is housed on a single server. We don't have a fine-grained access method for data sharing that lets us manage which people have access and which cannot, therefore we can't use granular techniques. The introduction of Attribute Based Encryption makes it possible for users to provide the names of other users as attributes; then, encryption is applied to all of those attributes, and only those users will be able to decode and access the data. Even with the implementation of Attribute Based Encryption (ABE), the data was still stored on a single, centralized server, where an attacker or administrator could access and change the data without any safeguards to prevent data privacy or immutability.

Distributed data storage (data will be stored at multiple nodes, so if one goes down, users can access services from other working nodes), data encryption, verification, and immutability (data once stored cannot be changed by attackers or admin users) are all features built into blockchain technology, which was introduced to avoid the data privacy and security issues mentioned above. Each record in a blockchain is stored as a block, and each block is associated with a unique hash code. When new data is added to the blockchain, it checks the hash codes of all the blocks to make sure the data is unmodified. If the hashes are the same, then the verification is successful. The hash algorithm and verification mechanism used by blockchain make it

considered unchangeable. The author of this research is motivated to secure data exchange by using Blockchain due to the aforementioned benefits. The distributed nature of blockchain storage makes it possible to cut down on unnecessary overhead.

We used five distinct approaches, detailed below, in their proposed work to handle data exchange safely.

1) IPFS: The Interplanetary File System (IPFS), which is a secure way to store encrypted information. IPFS stores files in internal memory and returns their addresses as hash codes.

2) Data Owner: The Data Owner, who is responsible for uploading and sharing files with other users.

3) Data Requester: There's the data requester, who makes a request to the data owner for access to certain files; the owner then generates Chebyshev polynomial encryption keys by XORing user information.

4) Smart Contract: We construct methods to store and obtain register user information, access control details, file hash code details, etc. in this smart contract, which is a tiny piece of code developed in SOLODITY programming that comprises functions linked to applications.

5) Blockchain: A decentralized database that stores and processes user information according to rules laid forth in smart contracts.

The seven distinct steps for completing the five modules mentioned above are as follows:

1) Setup Phase: Which the settings for the encryption keys will be defined.

2) Registration Phase: The data owner uses the smart contract access control mechanism to associate encryption keys with the required data.

3) Initialization Phase: The data owner will initialize all data requester lists.

4) Data Encryption and Storage: we encrypt the data using the keys we

produced before using polynomial XOR operations. We save the encrypted file in the IPFS system. After that, we get the hash code address of the file from IPFS. Finally, we deposit this hash code in the Blockchain. At any point in time when we require access to a file's decrypted contents, we may get its hash code from the blockchain and feed it into IPFS.

5) Authentication: this section will let you decide which users may access which files when you share them.

6) Testing Phase: The user's authorization to access the file will be hashed using the generated authentication

7) Access Control: The aforementioned authorization will be saved in the Blockchain SMART contract for use in the access control phase.

The next screen displays the code for a smart contract that can handle the access information mentioned before; this code is necessary to carry out the aforementioned operation.

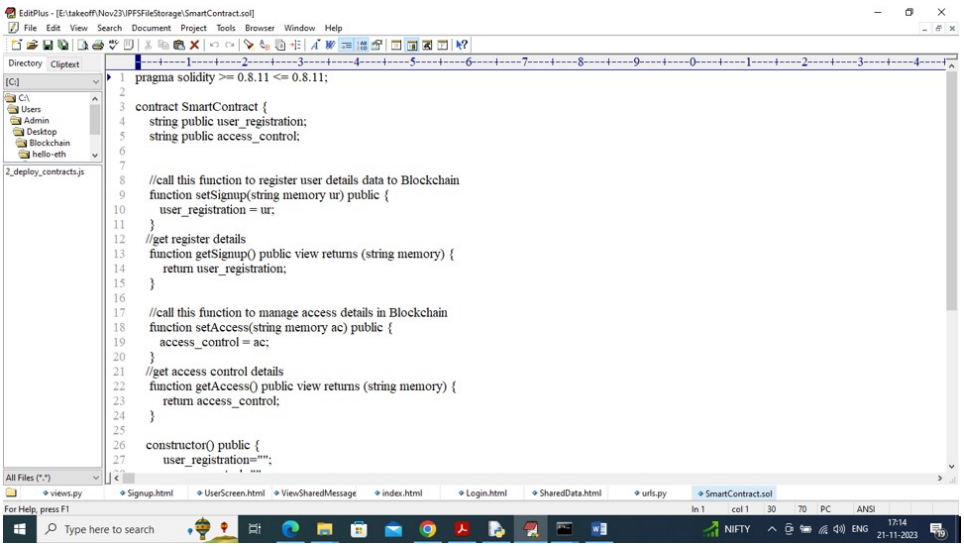


Figure 5.1: Smart Contract code.

Above smart contract code contains functions to manager user registration and access details and now we need to deploy above contract to Blockchain

Ethereum to store and get access details. To deploy contract we need to follow below steps

1) First go inside 'hello-eth/nodemodules/bin' folder and then look for file called 'runBlockchain.bat' and then double click on that file to get below screen

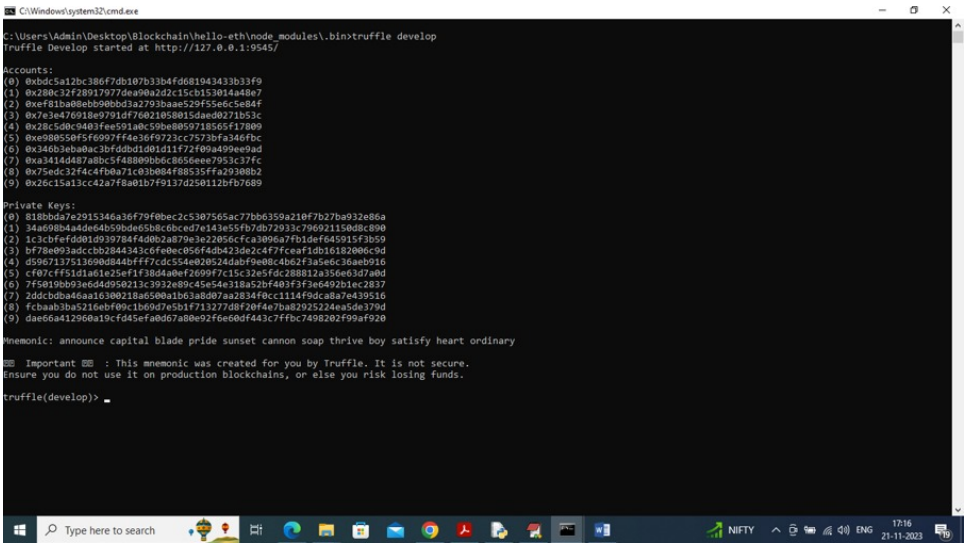


Figure 5.2: Blockchain Ethereum started.

2) In above screen Blockchain Ethereum started with default private keys and account address and in above screen just type command as 'migrate' and press enter key to deploy contract and get below page

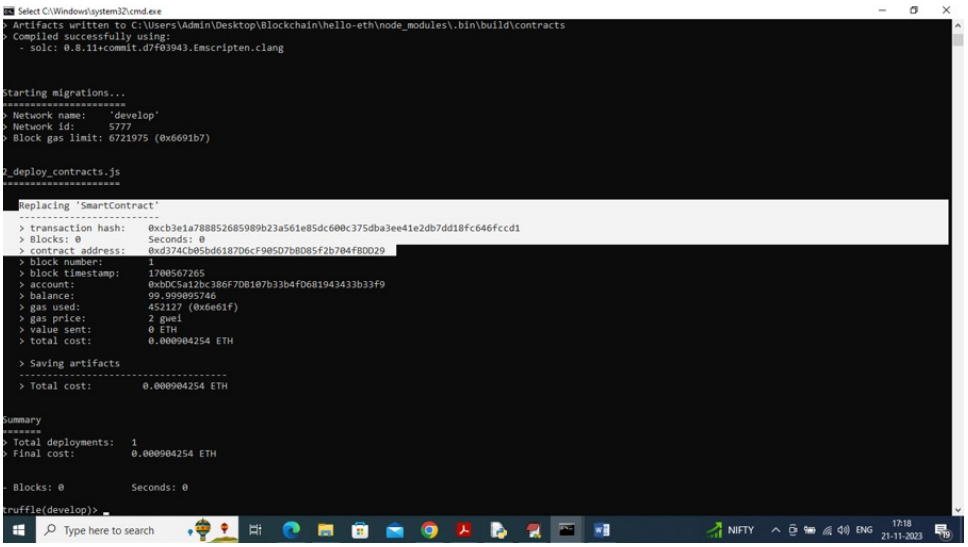


Figure 5.3: Smart Contract deployed.

3) In above screen in white colour text can see 'Smart Contract' deployed and we got contract address also and this address need to specify in python code to call contract to store and access sharing details. In below screen showing python code calling smart contract using address

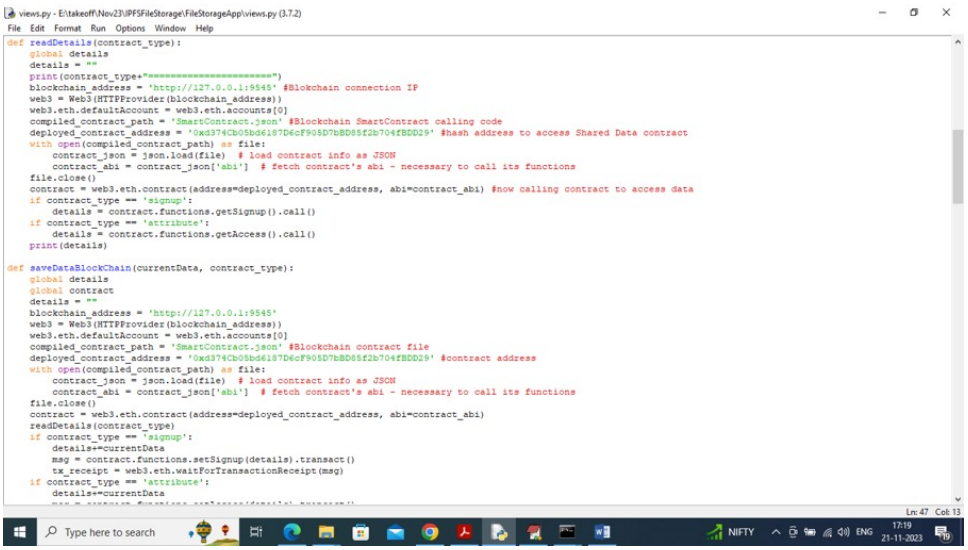


Figure 5.4: About contract calling.

In above screen read red colour comments to know about contract calling

Chapter 6

HARDWARE/ SOFTWARE TOOLS USED

PYTHON LANGUAGE:

Python is a high level object oriented and programming language with flexible syntax. Its flexible typing, variable binding, and high-level integrated data structures makes it ideal for scripting, quick application development, and connecting disparate components. Python's focus on accessibility and simple syntax reduces the cost of software maintenance. By supporting modules and packages, Python encourages code reuse and the modularization of programs. Both the Python interpreter as well as standard library are openly distributed and accessible on all major platforms, both in source code and binary format.The enhanced productivity that Python offers is a major selling point for many programmers. The edit-test-debug cycle is lightning quick since compilation is not involved. Python program debugging is a breeze since segmentation faults are never caused by bugs or improper input. On the contrary, If the interpreter identifies a mistake, an exception shall be granted. If the program fails to catch an exception, a stack trace is generated by the interpreter. A source level debugger allows you to travel through the code line by line, check variables both locally and globally, evaluate arbitrary expressions, create breakpoints, and do other tasks. The truth is that the debugger is developed in Python speaks much about the reflective capability of the language. However, putting some print statements within the source code is often a simple way to debug a program. This fundamental method works well because of its quick edit-test-debug cycle.

Python is a dynamic programming language that is interpreted, high-level, free, and open-source. It is suitable for both traditional and Object-Based Programming. Being a dynamically typed language, Python eliminates the need to define the type of variables. Take the equation x = 10 as an illustration. In this function, x may be any kind of String or int, etc.

Python Functionalities

Python is having numerous technology's some of them will be covered here:

1. Open Source and Free

If you want to download Python from the official website, you may do so for free by clicking on the Download Python option keyword down below. Install Python Being open-source also implies that anybody may access the program's source code. For your convenience, you may download, use, and share it.

2. GUI Programming Support

Graphical User Interface Programming Graphical Python modules like PyQt5, PyQt4, wxPython, or Tk may be used to create user interfaces. When it comes to building Python graphical applications, PyQt5 is by far the most popular choice.

3. Advanced Word Flow

A high-level language is Python. Python removes the requirement to memorize the system architecture and handle memory management when developing applications.

4. Adaptable characteristic

Python has the ability to be extended. We are able to translate Python code into C and C++, as well as build it in those languages.

5. Language Interpretation:

Python is classified as an interpreted language since it analyzes code line

by line. Since Python does not need compilation, debugging becomes more simpler, much like in C, C++, Java, etc. Bytecode is the immediate form of Python's source code.

6. Comprehensive Standard Library

You may avoid creating your own code for each item by making use of Python's huge standard library, which includes a wide range of modules and functions. Python has several libraries, including regular expressions, unit testing, web browsers, and much more.

7. Language that is dynamic type

Python is a programming language with dynamic typing. As a result, we do not need to define the variable's type (int, double, long, etc.) because it is determined at runtime rather than in advance.

8. Developing both the front end and the rear end

The use of basic HTML elements allows you to compile and design Python routines in a new project's py script. With this, you can use Python, just like javascript, to build front-end applications. Due to its many useful frameworks, such as Django and Flask, Python's backend is a popular choice for this kind of work.

9. Dynamic Memory Allocation

The data type of a variable is implicit in Python. At runtime, variables are automatically allocated memory as their values are assigned. If the integer value 15 is assigned to y, then developers won't have to write int y = 18. Enter y=18 if you so choose. .

Books and Software :-

ChaCha

ChaCha, a variation of Salsa, is a lightweight stream cipher suitable for IoT applications.ChaCha, like Salsa, is made up of 16 words in a 4×4 matrix, each carrying 32 bits. The 4×4 matrix is written as

$$S = \begin{bmatrix} s_0 & s_1 & s_2 & s_3 \\ s_4 & s_5 & s_6 & s_7 \\ s_8 & s_9 & s_{10} & s_{11} \\ s_{12} & s_{13} & s_{14} & s_{15} \end{bmatrix} = \begin{bmatrix} c_0 & c_1 & c_2 & c_3 \\ k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ t_0 & t_1 & v_0 & v_1 \end{bmatrix}$$

The first matrix $(s_0, s_1, s_2, \ldots, s_{1_5})$ represents the initial state words, while the second matrix (c_0, c_1, c_2, c_3) represents the preset constants, $k_0, k_1, \ldots, k_7$ represents the 256-bit key, and IV = (t_0, t_1, v_0, v_1) represents the 64-bit counter and 64-bit once.ChaCha uses the non-linear quarterround function, which differs from Salsa.Quarterround (w_1, x_1, y_1, z_1) is computed as

$w_1 = w_1 + x_1; z_1 = z_1 \bigoplus w_1; z_1 = z_1 <<< 16;$

$y_1 = y_1 + z_1; x_1 = x_1 \bigoplus y_1; x_1 = x_1 <<< 12;$

$w_1 = w_1 + x_1; z_1 = z_1 \bigoplus w_1; z_1 = z_1 <<< 8;$

$y_1 = y_1 + z_1; x_1 = x_1 \bigoplus y_1; x_1 = x_1 <<< 7;$

In contrast to Salsa, ChaCha employs both the column and diagonal rounds. Even rounds have been referred to as diagonal rounds, and odd rounds as column rounds.To compute column-round (S) and diagonal-round (S), start with a 16-word input matrix $(s_0, s_1, \ldots, s_{1_5})$.

Column-round (S):

quarterround (s_0, s_4, s_8, s_{12})

quarterround (s_1, s_5, s_9, s_{13})

quarterround $(s_2, s_6, s_{10}, s_{14})$

quarterround $(s_3, s_7, s_{11}, s_{15})$

Diagonal-round (S):

quarterround $(s_0, s_5, s_{10}, s_{15})$

quarterround $(s_1, s_6, s_{11}, s_{12})$

quarterround (s_2, s_7, s_8, s_{13})

quarterround (s_3, s_4, s_9, s_{14})

Each round of ChaCha is reversible and is referred to as the reverse round of ChaCha. Reverse-round (w_1, x_1, y_1, z_1) is computed as

$x_1 = x_1 >>> 7; x_1 = x_1 \bigoplus y_1; y_1 = y_1 z_1;$

$z_1 = z_1 >>> 8; z_1 = z_1 \bigoplus w_1; w_1 = w_1 x_1;$

$x_1 = x_1 >>> 12; x_1 = x_1 \bigoplus y_1; y_1 = y_1 z_1;$

$z_1 = z_1 >>> 16; z_1 = z_1 \bigoplus w_1; w_1 = w_1 x_1;$

Tensorflow

The open-source dataflow and differentiable programming framework TensorFlow is available for free and may be used for many different kinds of projects. Machine learning tools like neural networks make use of this symbolic math library. Google use it for both internal research and external manufacturing.

Google's Brain Team has developed TensorFlow for use in the workplace. It was made available under the Apache 2.0 free and open source license on November 9, 2015.

Stupid Puppy

An array-processing package with general-purpose capabilities is Numpy. It offers a high-performance object for multidimensional arrays and tools to manipulate them.

Python scientific computing is a foundational package.Which is having several characteristics, the most essential of which are: - an efficient N-dimensional array object. - Advanced streaming functions. - Resources for combining C/C++ and Fortran code - Practical experience with linear algebra, Fourier transform, and random number generation.

Numpy has several obvious scientific uses, but it also makes a great multidimensional data container. Because it allows you to define any kind of data type, Numpy can connect database easily to a broad number.

Wild pandas

Pandas is a powerful tool for handling and evaluating data due to its robust data structures. Python was mostly utilized to perform data munging and processing. In terms of analyzing the data, it was generally unsuccessful. The pandas worked it out.Pandas covers the five most common processing and evaluation steps, regardless of data source: load, organize, edit, model, and evaluate. Python and Pandas are used in a wide range of academic as well as professional disciplines, including economics, statistics, analytics, finance, and many more.

Use Matplotlib

Whether you're working in an interactive environment or working with hardcopy formats, Matplotlib, a Python 2D plotting toolkit, can generate publication-quality figures. You may use Matplotlib in a variety of places: scripts developed in the programming language Python, including the Python, the IPython shells, the Jupyter Notebook and web application servers are the four GUI resources. Matplotlib's purpose is to simplify difficult tasks while enabling simple ones. Plots, histograms, power spectra, bar charts, error charts, scatter plots, and other graphics may be created with minimal code. See the thumbnail galleries and specimen charts for

explanations.

The pyplot component, when integrated with IPython, provides a MATLAB-like interface, making it excellent for simple charting jobs.Users of having complete power control over line styles, font quality, axis attributes, and other factors, whether using an object-oriented interface or ways known to MATLAB users.

Discover Scikit-360

Scikit-learn's basic Python interface supports a wide range of supervised and unsupervised learning techniques. Academic and business institutions are encouraged to utilize it since it is offered under numerous Linux versions and has a liberal simplified BSD license.

Chapter 7

SYSTEM TESTING & RESULTS

7.1 SYSTEM TESTING

Quality assurance (QA) teams do system testing, which is sometimes called system-level testing or system-integration testing, to determine how an application's many parts work together as a whole. The purpose of system testing is to ensure that an application works as expected. This stage is similar to black box testing in that it examines the app's inner workings. As an example, system testing may check all user input types consistently provide the expected results across the program.

Steps in testing a system: A video guide to this level of the exam. The goal of system testing is to ensure that an application's many parts function together smoothly. System analys is the last step in the quality assurance process, which usually begins with functional or user-story testing on individual modules and continues with integration testing on each component. Following successful completion of system testing, a software build is subjected to acceptance testing as a last check before being released to production, where it is used by actual users. Every issue is documented by the app-dev team, which also decides how many and what sorts of bugs are acceptable.

7.1.1 METHODS FOR TESTING SOFTWARE

For software engineering testing to be successful, the technique must be optimized. A software testing plan specifies the when, what, and how of ensuring a high-quality final result. In most cases, this overarching goal is accomplished via the employment of the following software testing strategies:

Functional Testing:

Static testing, in which the developing product is not run, is the early-stage testing technique. Bugs and flaws in the code itself can only be found by desk-checking. To prevent issues brought on by bugs in the code or deficiencies in the software's structure, it is crucial to do such a check-up before deployment.

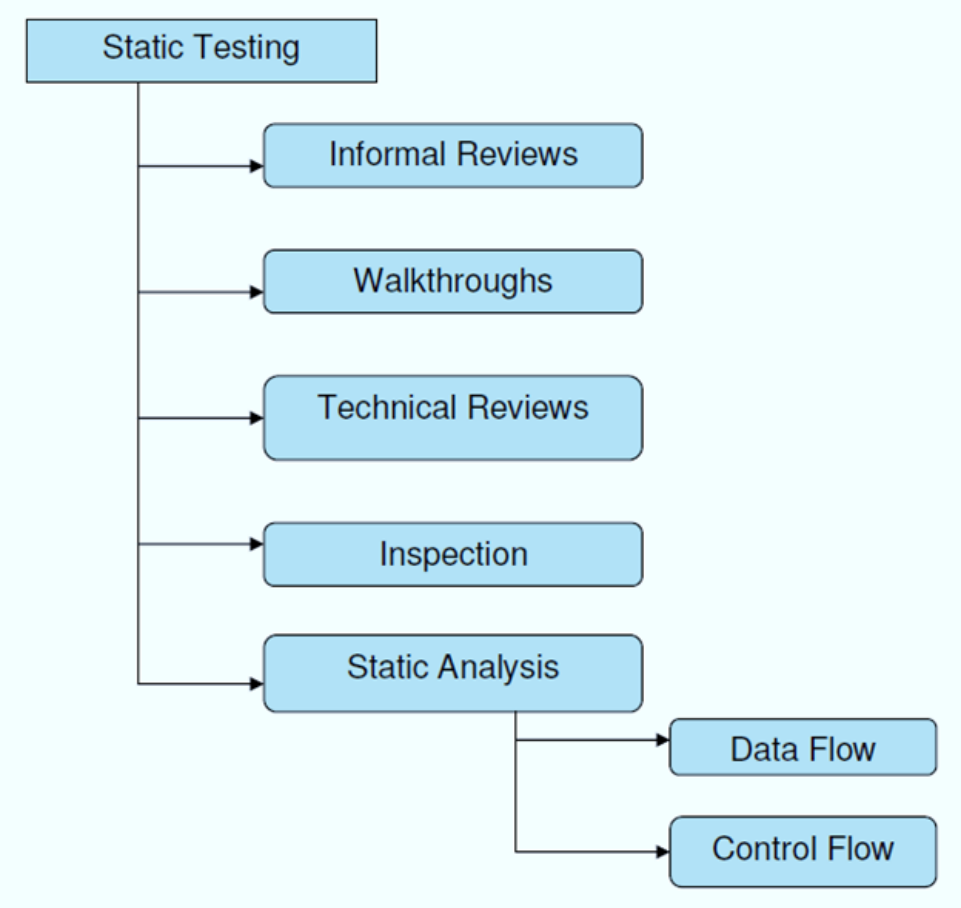


Figure 7.1: Static Testing.

Structural Testing Without actually executing the program, proper testing cannot be done. Finding and fixing mistakes and errors that arise in the starting stage of software improvement requires structural testing, which is also called white-box testing. In this case, a regression test is used to perform unit tests based on the structure of the program. Typically, this step of development is automated and carried out inside the test automation framework. The software structure and data flows are fully accessible by developers and quality assurance engineers in the Data Flow test. It enables them to monitor changes in the system's behaviour through mutation tests that compare test results with those of earlier versions.

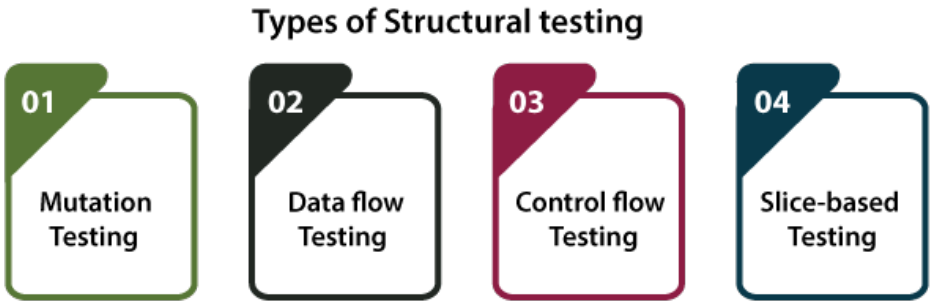


Figure 7.2: Structural Testing.

Behavioral Testing Instead of delving into the inner workings of the program, this last round of testing focuses on how it responds to certain actions. Behavioral testing, sometimes called black-box testing, involves observing the product from the user's perspective via a battery of tests, the majority of then are manual. In a way to do usability tests, for instance, and respond to issues in the same way that normal users of the material would, QA engineers often have particular knowledge about a company or software goals ('the black box'). If repeated tasks are necessary for behavioral testing, automation (regression tests) may be used to remove human mistake. For instance, to gauge the product's performance under heavy load, you could have to fill out 100 online registration forms; hence, automating this test would be ideal.

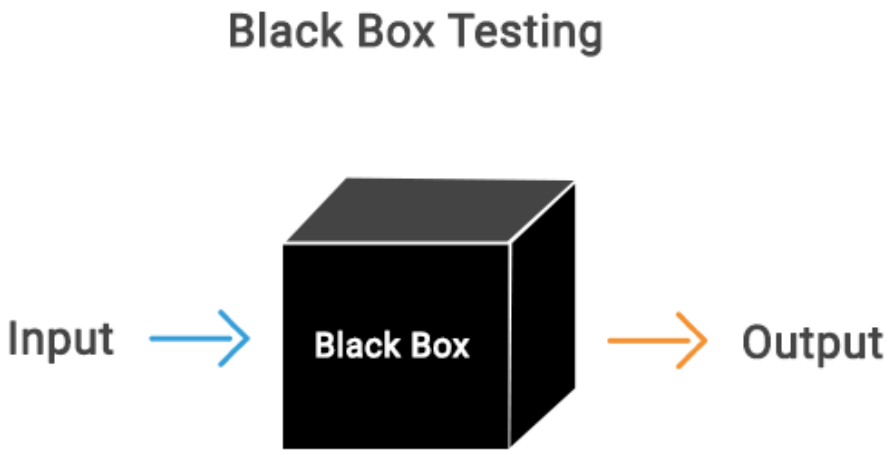


Figure 7.3: Black box Testing.

Table 7.1: Test Cases

S.NO	INPUT	If available	If not available
1	Shared Secured Data	write some description and then upload file	There is no process
2	View Shared Messages	data requester can view all details	There is no process
3	Storage Overhead Graph	graph x-axis represents ABE and propose algorithms and y-axis represents communication over or execution time and in both algorithms	There is no process
4	View Shared Messages	access permission so he can view or download file.	There is no process

7.2 RESULTS

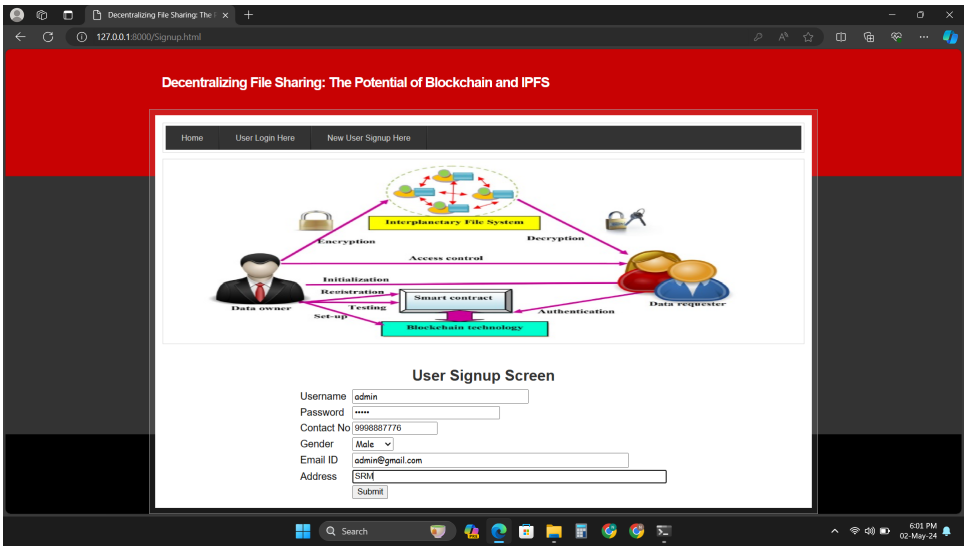


Figure 7.4: User registration.

In the figure 7.4 user is registering as ‘admin’ and later sign up will be in the figure 7.5

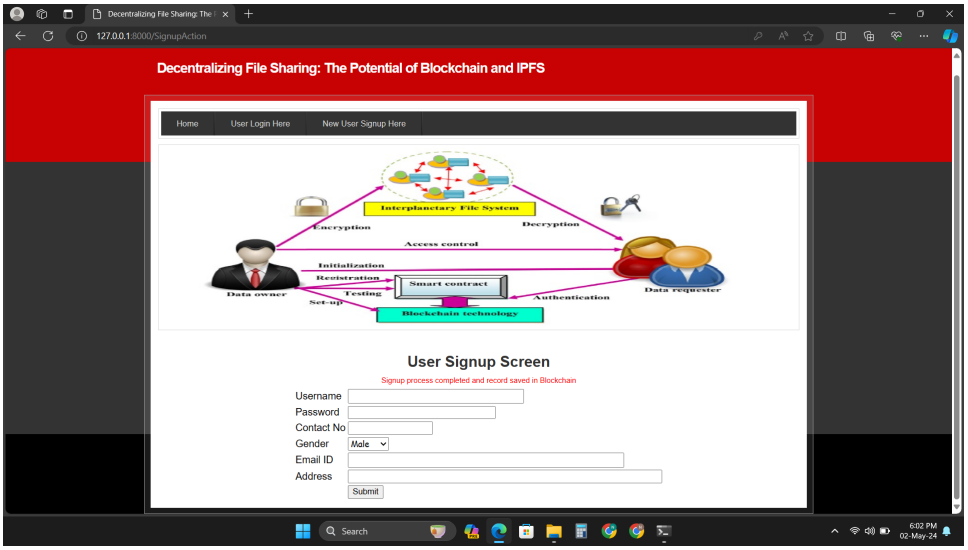


Figure 7.5: Successfull registration.

In the figure7.5 we can see sign up completed and similarly add another user

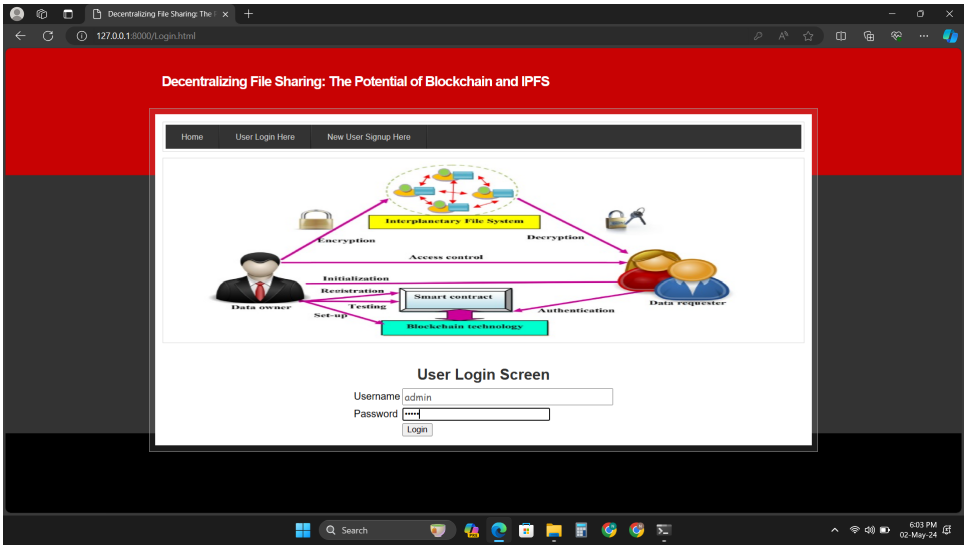


Figure 7.6: User login page.

In the figure7.6 user is login as ‘admin’ and after login will get below page.

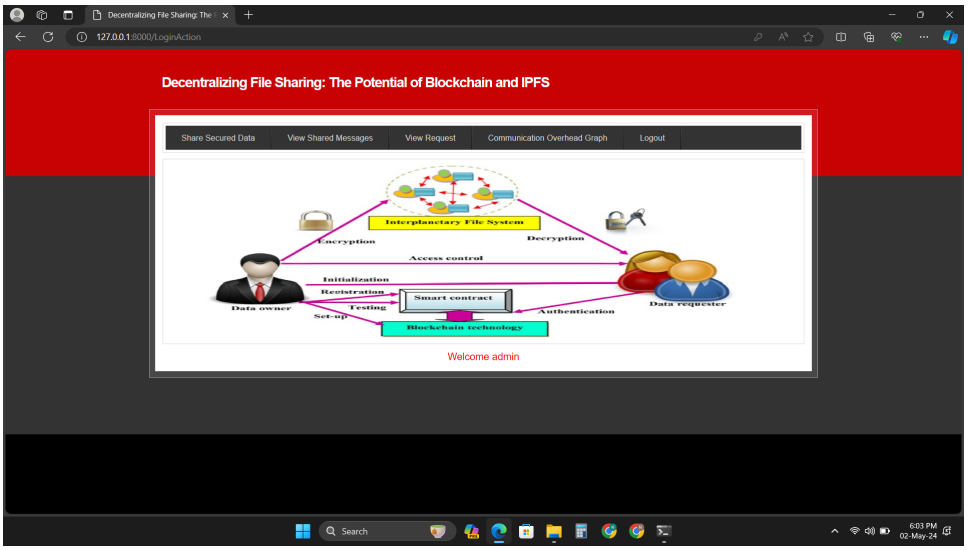


Figure 7.7: Landing page.

In the figure7.7 user can click on ‘Shared Secured Data’ link to get below page.

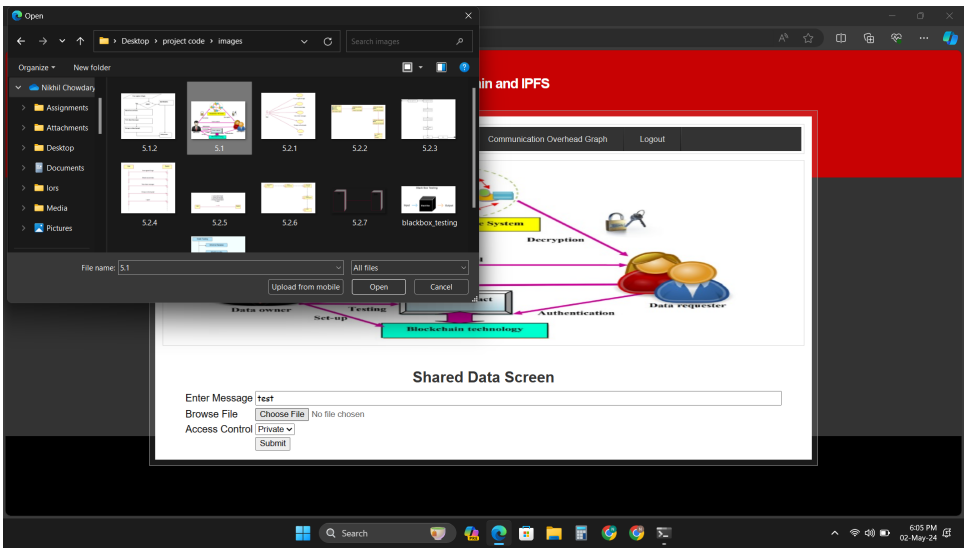


Figure 7.8: Uploading as Private.

In above screen uploading one file with access control as ‘Private’ and then press ’Submit’.

Now similarly uploading another file Public option.

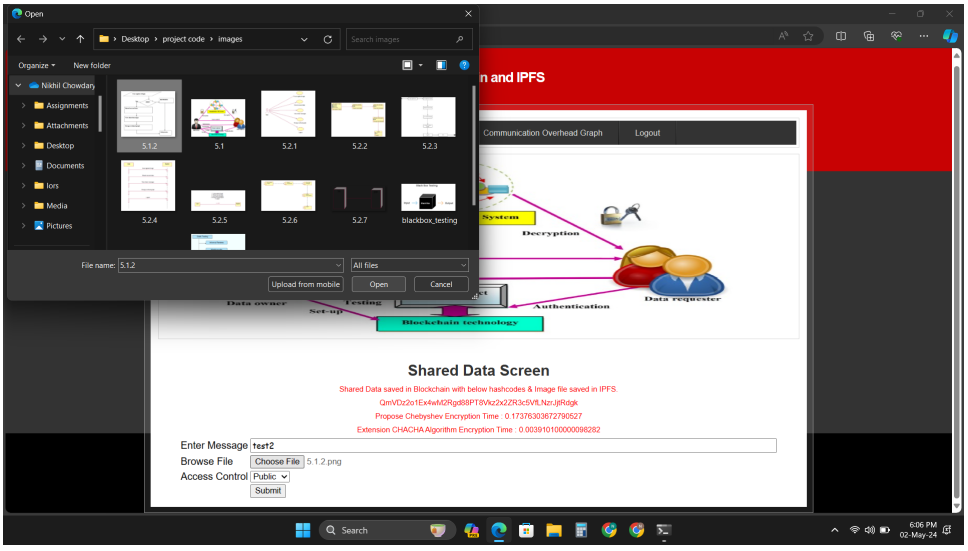


Figure 7.9: Uploading as Public.

In above screen uploading another file with access as ‘Public’ and

press ‘Submit’ button.

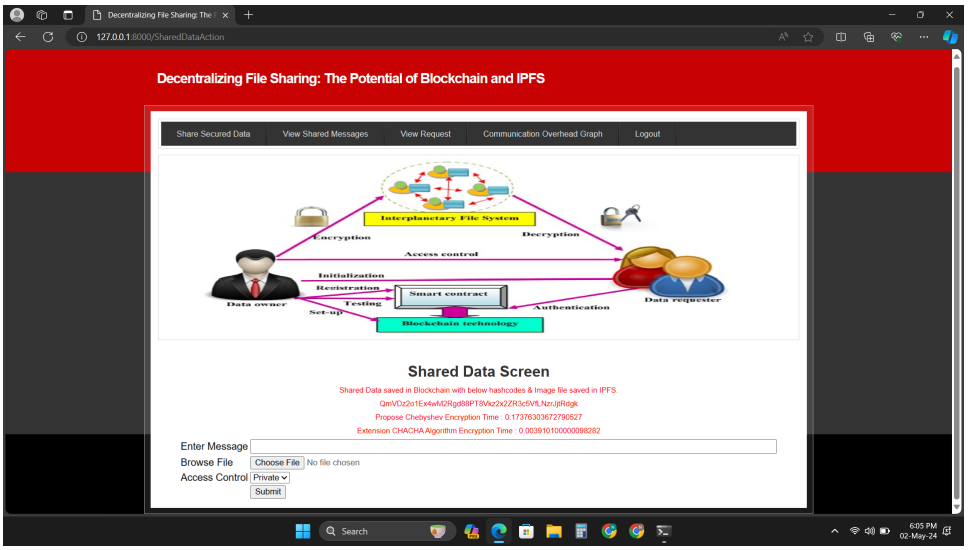


Figure 7.10: After uploading a file.

In above screen file details saved in Blockchain and IPFS.

In above screen another file also saved and now this owner can click on ‘View Shared Messages’ link to get below page.

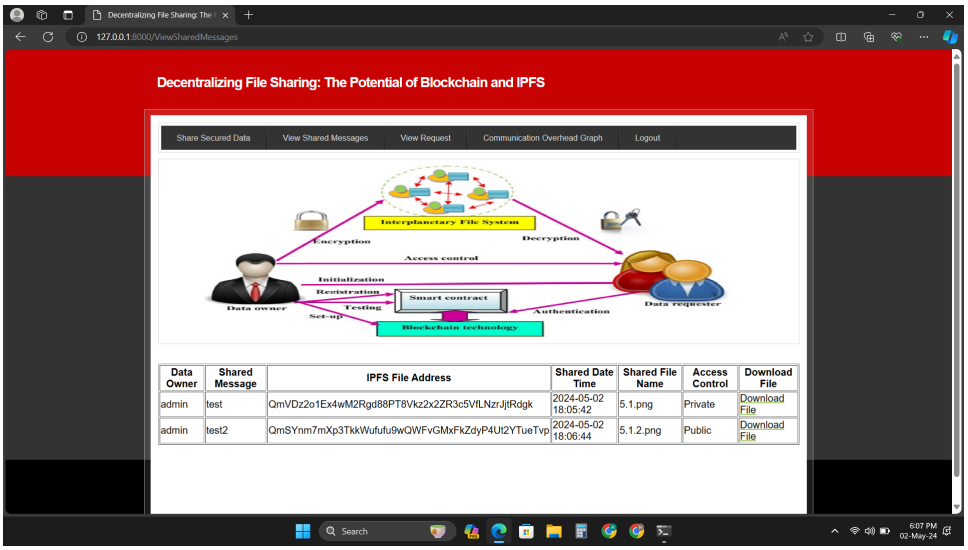


Figure 7.11: View shared messages.

In above screen as this is owner so by default he can access all files and now logout and login as another user ‘user’ to perform permission

operations

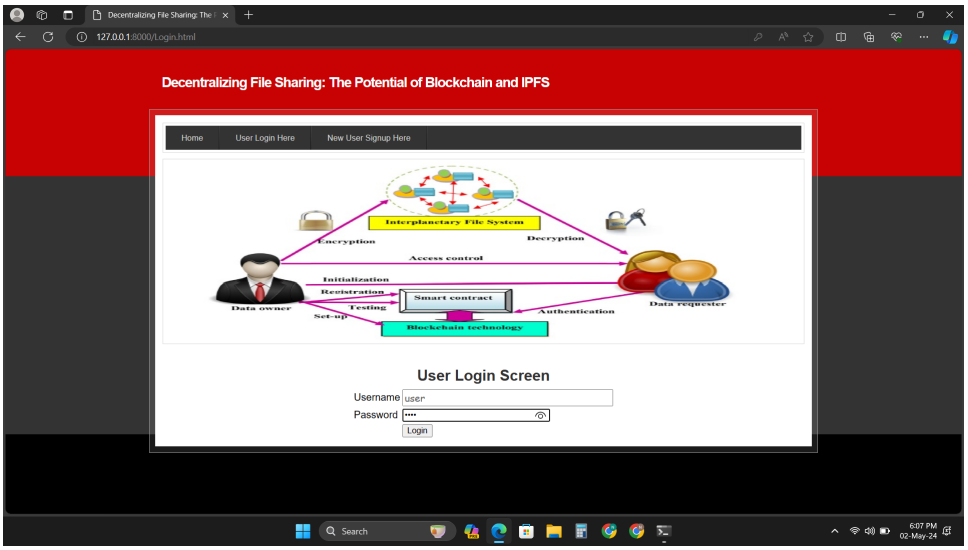


Figure 7.12: User login.

In above screen ‘user’ user is login and after login will get landing page.

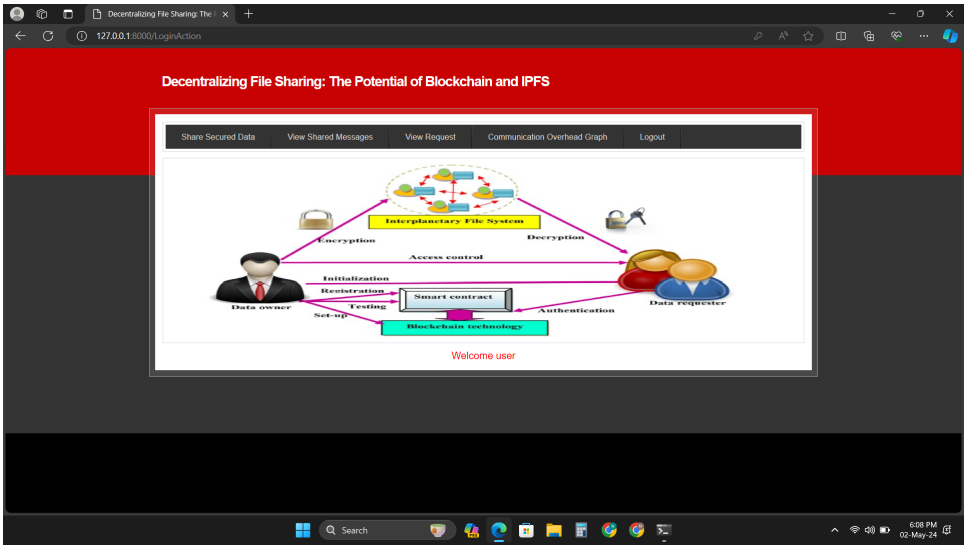
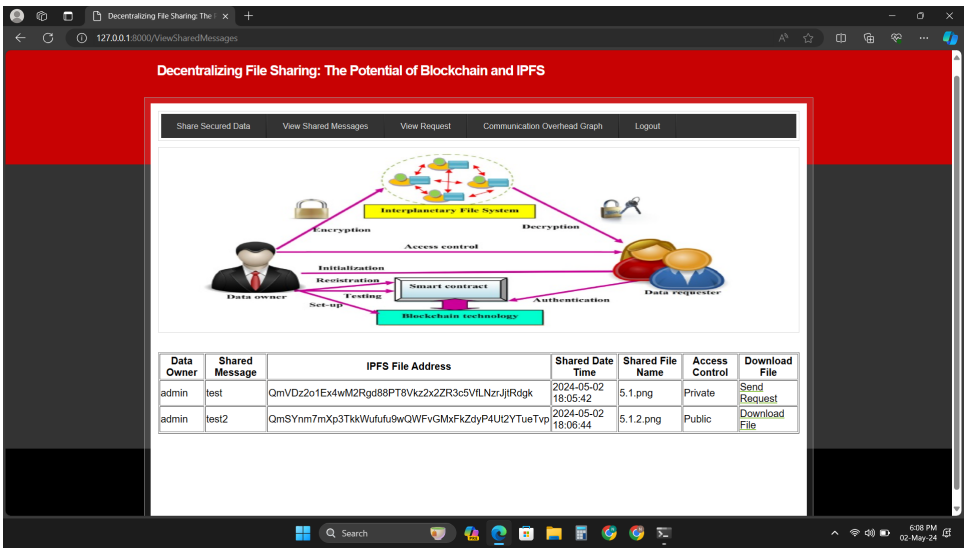


Figure 7.13: Home page of user.

In above screen user ‘user’ can click on ‘View Shared Messages’ link to get below page.



Data Owner	Shared Message	IPFS File Address	Shared Date Time	Shared File Name	Access Control	Download File
admin	test	QmVDz2o1Ex4wM2Rgd88PT8Vkz2x2ZR3c5VfLNzrJjtRdgk	2024-05-02 18:05:42	5.1.png	Private	Send Request
admin	test2	QmSYnm7mXp3TkkWufufu9wQWFvGMxFkZdyP4Ut2YTueTvp	2024-05-02 18:06:44	5.1.2.png	Public	Download File

Figure 7.14: Messages availble.

In above file list we can see the file with ‘Public’ access is available freely for download and then file with ‘Private’ access has a link as ‘Send Request’ and now user can click on ‘Send Request’ link to send request to owner and get below page.

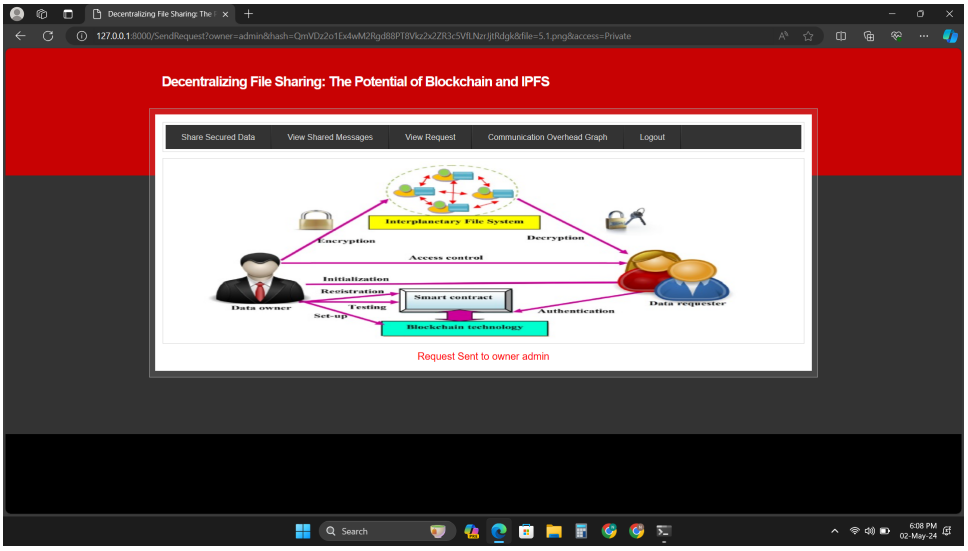


Figure 7.15: User screen after successfully placing request.

In above screen in text can see ‘request sent to owner admin’ and now logout and login as owner ‘admin’ to perform ‘read or download’ permission.

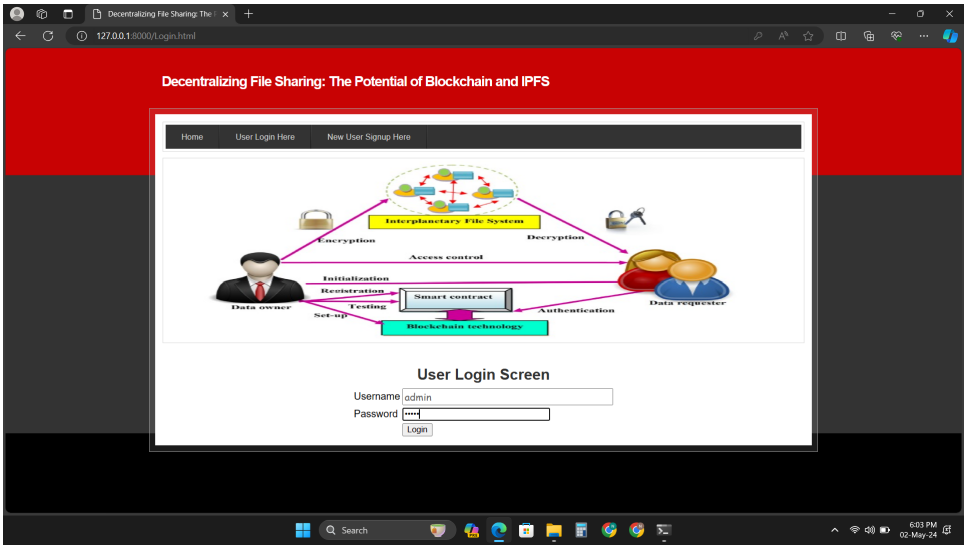


Figure 7.16: Logging in as admin

In the figure 7.16 user ‘admin’ can login and then after login moves to the landing page click on ‘View Request’ link to view all requests from other user.

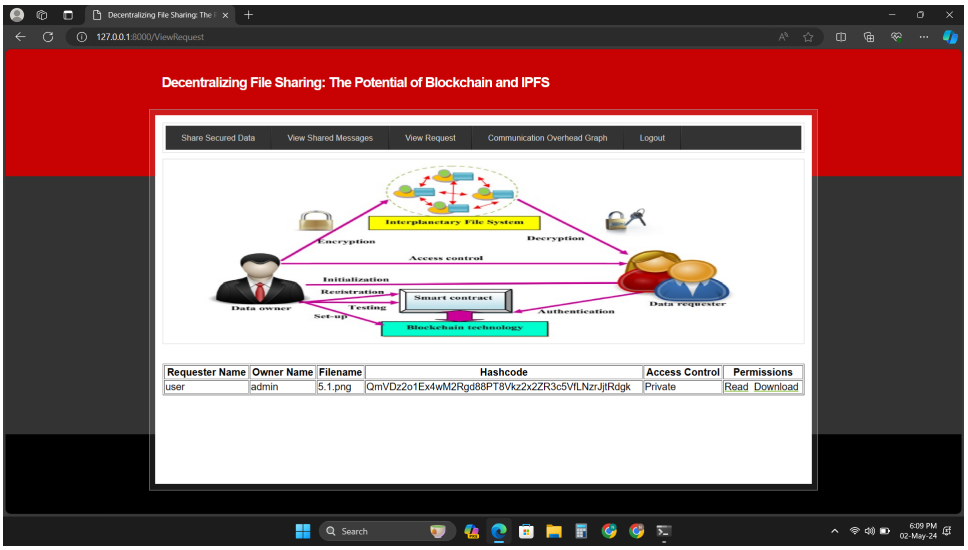


Figure 7.17: Request page.

In the figure7.17 owner can find all file requests and now check on either ‘Read or download’ link to provide desired permissions and then will get below page.

Figure 7.18: Change of status.

In above screen I gave file permission as ‘read’ and now logout and login as user ‘user’ so he can access file with read option.

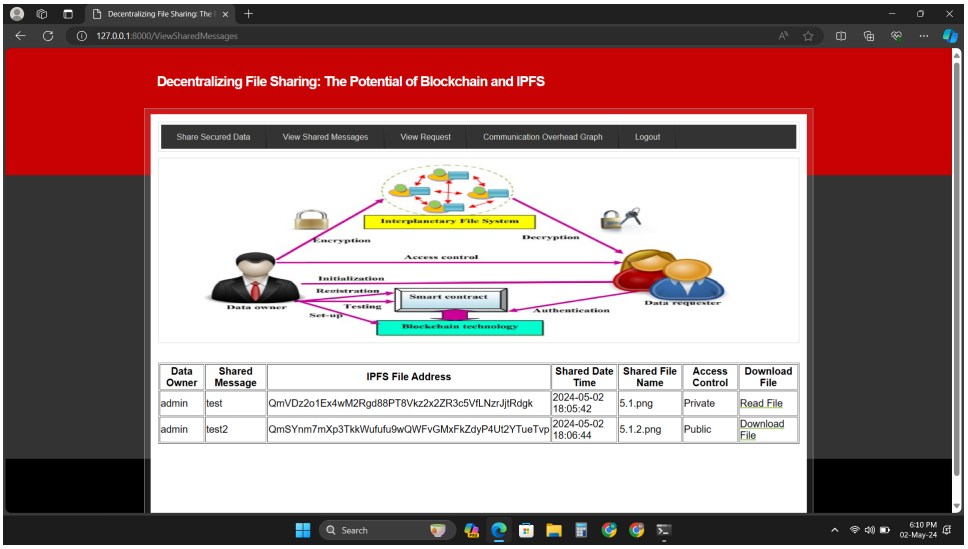


Figure 7.19: User shared messages page.

In above screen can see private file ‘option’ changed to ‘Read File’ and then click on that link to read the file content like below screen.

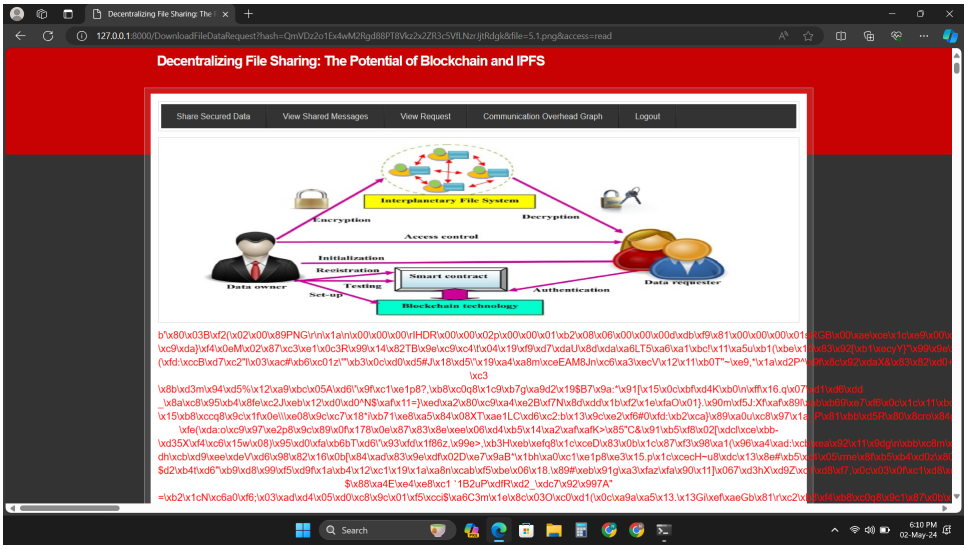


Figure 7.20: Opening file with reading mode.

In the figure7.20 user can find the file data in reading format and similarly can click on ‘Download File’ to download the file.

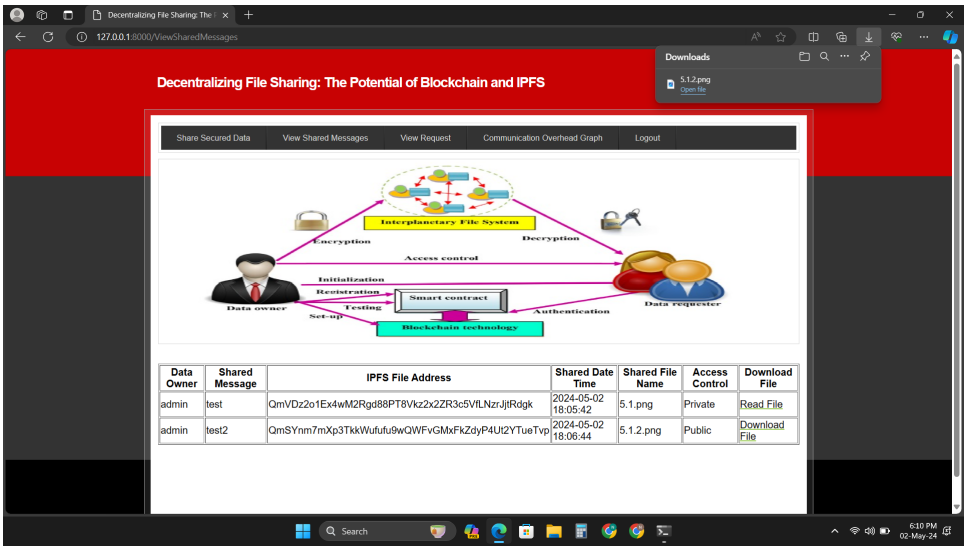


Figure 7.21: Downloading file.

In above screen after click in ‘Download’ can see file is downloaded.

In propose work author has used 'Chebyshev' algorithm for encryption which is little high in computation so as extension we have employed CHACHA20 encryption algorithm which is 70% faster and more secured than propose algorithm.While uploading a file we can see that on screen for file uploading and encryption we have applied both algorithms and then recorded encryption time for both propose and extension algorithms.

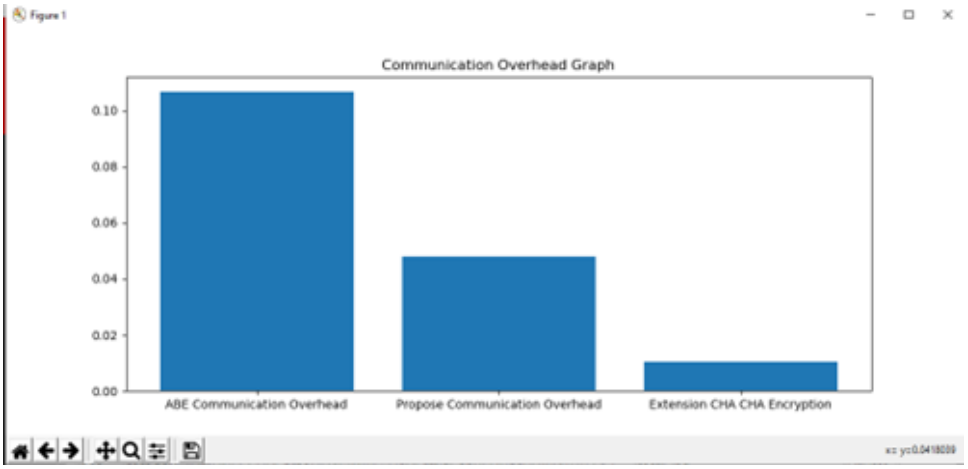


Figure 7.22: Comparison between algorithms.

In the figure 7.22 x-axis represents the names of algorithms which used and the y-axis represents the Computation Time and in all algorithms Extension CHACHA20 is more efficient.

Chapter 8

CONCLUSION

When it comes to online data sharing settings, centralized servers have their limitations. The suggested approach offers a strong alternative. The solution guarantees improved security, privacy, and resilience against unwanted access and data breaches by using Blockchain, IPFS, and Attribute-Based Encryption (ABE). Users may safely access and exchange data with granular access control thanks to the integration of these technologies, which reduces the risks linked to vulnerabilities in centralized servers. While ABE allows for granular access restriction based on user traits, IPFS's decentralized nature makes dispersed data storage possible. Implementing smart contracts on the blockchain further strengthens the system's stability and integrity by managing application functionalities. The suggested work expands upon previous efforts by substituting the computationally demanding Chebyshev algorithm with the faster and more secure CHACHA20 method, which increases speed and dependability by 70%. Users may rest certain that their shared data will remain secret, intact, and readily available thanks to the proposed system's all-encompassing approach to safe data sharing. To make the system even more useful in actual deployments, future updates may look at ways to make it more efficient and scalable.

REFERENCES

[1] **Q. Chen, G. Srivastava, R. M. Parizi, M. Aloqaily, and I. A. Ridhawi,** "An incentive-aware blockchain-based solution for Internet of fake media things," Inf. Process. Manage., vol. 57, no. 6, Nov. 2020, Art. no. 102370.

[2] **H. R. Hasan and K. Salah,** "Blockchain-based solution for proof of delivery of physical assets," in Blockchain–ICBC, S. Chen, H. Wang, and L.-J. Zhang, Eds. Cham, Switzerland: Springer, 2018, pp. 139–152.

[3] **K. Salah, N. Nizamuddin, R. Jayaraman, and M. Omar,**"Blockchain-based soybean traceability in agricultural supply chain," IEEE Access, vol. 7, pp. 73295–73305, 2019.

[4] **A. Yousafzai and C. Seon Hong,** "SmartSON:A smart contract driven incentive management framework for self-organizing networks," 2020, arXiv:2008.11803. [Online].

[5] **A. Ouaddah, A. Abou Elkalam, and A. Ait Ouahman,** "FairAccess: A new blockchain-based access control framework for the Internet of things," Secur. Commun. Netw., vol. 9, no. 18, pp. 5943–5964, Dec. 2016.

[6] **D. Tith, J.-S. Lee, H. Suzuki, W. Wijesundara, N. Taira, T. Obi, and N. Ohyama,** "Application of blockchain to maintaining patient records in electronic health record for enhanced privacy, scalability, and availability," Healthcare Inform. Res., vol. 26, no. 1, pp. 3–12, 2020.

[7] **M. M. Payeras-Capella, M. Mut-Puigserver, and M. A. Cabot-Nadal,** "Blockchain-based system for multiparty electronic registered delivery services," IEEE Access, vol. 7, pp. 95825–95843, 2019.

[8] **N. Z. Aitzhan and D. Svetinovic,** "Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams," IEEE Trans. Dependable Secure Comput., vol. 15, no. 5, pp. 840–852, Sep. 2018.

[9] **H. Al Breiki, L. Al Qassem, K. Salah, M. Habib Ur Rehman, and D. Sevtinovic,** "Decentralized access control for IoT data using blockchain and trusted oracles," in Proc. IEEE Int. Conf. Ind. Internet (ICII), Nov. 2019, pp. 248–257.

[10] **H. Hu, G.-J. Ahn, and J. Jorgensen,** "Multiparty access control for online social networks: Model and mechanisms," IEEE Trans. Knowl. Data Eng., vol. 25, no. 7, pp. 1614–1627, Jul. 2013.

[11] **P. Ilia, B. Carminati, E. Ferrari, P. Fragopoulou, and S. Ioannidis,** P. Ilia, B. Carminati, E. Ferrari, P. Fragopoulou, and S. Ioannidis,

[12] **H. Guo, E. Meamari, and C.-C. Shen,** "Multi-authority attribute-based access control with smart contract," in Proc. Int. Conf. Blockchain Technol. (ICBCT), 2019, pp. 6–11.

[13] **C. Chinchilla. (2019).** A Next-Generation Smart Contract and Decentralized Application Platform. Accessed: Apr. 23, 2020. [Online].

[14] **D. Bryson, D. Penny, D. Goldberg, and G. Serrao,** Blockchain Technology for Government. Montgomery, AL, USA: The MITRE Corp., 2017.

[15] **(2018). Quorum Whitepaper. Accessed: Mar. 23, 2020. [Online].**

[16] **(2020). Besu Enterprise Ethereum Client. Accessed: Mar. 23, 2020. [Online].**

[17] **J. Benet. (2014).** IPFS–Content Addressed, Versioned, P2P File System. Accessed: Mar. 23, 2020. [Online].

[18] **Remix Docs Description. Accessed: Apr. 3, 2020. [Online].**

[19] **K. Hoffman, D. Zage, and C. Nita-Rotaru,** "A survey of attack and defense techniques for reputation systems," ACM Comput. Surv., vol. 42, no. 1, pp. 1–31, Dec. 2009, doi: 10.1145/1592451.1592452.

[20] **Kakumani K.C. Deepthi , Kunwar Singh**"Cryptanalysis for reduced round Salsa and ChaCha: revisited,"IET Information Security,01 November, 2019.

[21] **Ammar Ayman Battah; Mohammad Moussa Madine; Hamad Alzaabi; Ibrar Yaqoob,** "Blockchain-Based Multi-Party Authorization for Accessing IPFS Encrypted Data",published in IEEE Access.

[22] **Jin Sun; Xiaomin Yao; Shangping Wang; Ying Wu,** "Non-Repudiation Storage and Access Control Scheme of Insurance Data Based on Blockchain in IPFS", published in IEEE Access.

[23] **Jin Sun; Xiaomin Yao; Shangping Wang; Ying Wu,** "Blockchain-Based Secure Storage and Access Scheme For Electronic Medical Records in IPFS", published in IEEE Access.

[24] **Muqaddas Naz, F. Al-Zahrani, Rabiya Khalid, N. Javaid, A. M. Qamar, M. Afzal, M. Shafiq,**"A Secure Data Sharing Platform Using

Blockchain and Interplanetary File System", published in Sustainability 10 December 2019, Computer Science, Engineering.

[25] **Xiaohong Zhang; Xiaofeng Chen,**"Data Security Sharing and Storage Based on a Consortium Blockchain in a Vehicular Ad-hoc Network", published in IEEE Access.

[26] **M. Aloqaily, A. Boukerche, O. Bouachir, F. Khalid, and S. Jangsher,**"An energy trade framework using smart contracts: Overview and challenges," IEEE Netw., vol. 34, no. 4, pp. 119–125, Jul. 2020.