

IMPROVING THE EFFICIENCY OF E-HEALTH CARE SYSTEM USING CLOUD COMPUTING AND BLOCKCHAIN

Submitted in partial fulfilment of the requirements for the award of
Bachelor of Technology degree in
Computer Science Engineering.

by

CH.LAKSHMI SARVANI (Reg no.37110122)

B.LIKHITHA SAGARI (Regno.37110073)



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

**(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600 119**

APRIL – 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **CH.LAKSHMI SARVANI** (Reg. No. **37110122**), **B.LIKHITHA SAGARI** (Reg. No. **37110073**) who carried out the project entitled "**Improving the efficiency of E- Health Care system using Cloud computing and Block Chain**" under my supervision from **September 2020 to March 2021**.

Dr.S.PRINCE MARY M.E., PH.D.,

Internal Guide

Head of the Department

Submitted for Viva voce Examination held on :

Internal Examiner

External Examiner

DECLARATION

I, **CH.LAKSHMI SARVANI (37110122)** hereby declare that the Project Report entitled “**Improving the Efficiency of E-Healthcare system using Cloud Computing and Block Chain**” done by us under the guidance of **Dr. S. Prince Mary M.E., Ph.D., Department of Computer Science Engineering** is submitted in partial fulfillment of the requirements for the award of **Bachelor of Technology** degree in **Computer Science Engineering**.

DATE:

CH.L.SARVANI

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We would like to express our sincere and deep sense of gratitude to our Project Guide **DR.S.PRINCE MARY M.E., Ph.D., Department of Computer Science and Engineering** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We convey our thanks to **Dr.T.SASIKALA, M.E., Ph.D., Dean, School of computing, Dr.S.VIGNESHWARI M.E., Ph.D., and Dr. LAKSHMANAN, M.E., Ph.D.,** Heads of the Department, **Department of Computer Science and Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We wish to express our thanks to all **Teaching and Non-teaching** staff members of the **Department of Computer Science & Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Electronic health records possess the patient's medication details and their health history. The health records attract the attention of the attackers' as it possesses invaluable information. Loss of electronic health record leads to a wrong medication or surgery. Healthcare systems provide fewer security measures to secure the health records. Blockchain is a distributed and decentralized ledger that plays a vital role in securing the data and transactions. Introduction of blockchain in the healthcare systems protects the health records from the attackers. However, the blockchain faces phishing, dictionary based, cold wallets, and hot wallets attacks. This paper proposes a multilevel authentication-based scheme to protect the blockchain from the attacks. Since the patient have no control over the data, the chances of data being misused is high. So we need a patient- centered approach which is completely decentralised, which can identify data thefts, prevent data manipulation, and patient has the right in access control. Blockchain Technology serves as a best solution to address all the problems and fulfil the needs. Blockchain being a decentralised and distributed ledger it can also impact on billing, record sharing, medical research, identify thefts and financial data crimes in days to come. Implementation of smart contracts in health care can simplify things even better. Where invoking, record creation and validation will be done on Blockchain.

TABLE OF CONTENTS

CHAPTERNo.	TITLE	PAGENO.
	ABSTRACT	v
	LIST OF ABBREVIATIONS	viii
	LIST OF ABBREVIATIONS	ix
1	INTRODUCTION	1
	1.1 OVERVIEW	1
	1.2 DOMAIN OVERVIEW	1
	1.2.1 Block chain	2
	1.2.2 Components of a block	3
	1.2.3 Proof of work	8
2	LITERATURE SURVEY	12
	2.1 LITERATURE SURVEY 1	12
	2.2 LITERATURE SURVEY 2	14
	2.3 LITERATURE SURVEY 3	17
3	AIM & SCOPE	18
	3.1 AIM OF THE PROJECT	18
	3.2 SCOPE OF THE PROJECT	18
4	METHODS AND ALGORITHMS USED	19
	4.1 EXISTING SYSTEM	21
	4.2 DRAWBACKS OF EXISTING SYSTEM	22
	4.3 PROPOSED SYSTEM	22
	4.4 ADVANTAGES OF PROPOSED SYSTEM	23

	4.5 LIST OF MODULES	24
	4.5.1 Module 1	24
	4.5.2 Module 2	25
	4.5.3 Module 3	26
5	RESULTS AND DISCUSSION	31
	5.1 LOGIN PAGE	31
	5.2 DASHBOARD	31
	5.3 CREATE EHR RECORD	32
	5.4 CLOUD AUDITOR	32
	5.5 AUDIT LISTING	33
	5.6 BLOCK CHAIN REPORT	33
6	SUMMARY AND CONCLUSION	34
	6.1 CONCLUSION	34
	REFERENCES	35
	APPENDIX	37

LIST OF FIGURES

FigNo.	TITLE	PageNo.
1.1	Hash Diagram	5
1.2	Network	6
4.1	RAD Model	20
4.2	Architecture Diagram	29

LIST OF ABBREVIATIONS

EHR	-	Electronic Health Record
DHT	-	Distributed hash table
DES	-	Data Encryption Standard algorithm
RSA	-	Rivert-shamir-Adleman algorithm
TPA	-	Third Party Auditor
CSP	-	Cloud Service Provider
RAD	-	Rapid Application Development Model

CHAPTER 1

INTRODUCTION

Along with all the advantages of cloud computing there comes a threat of security of sensitive outsourced data. Cloud users cannot rely only on cloud service provider for the security reasons of the outsourced data. On that account there is need of Third Party Authenticator who will authenticate the cloud data on behalf of data owners. This paper proposes two schemes in order to ensure the security of the data stored on cloud. First one is Public Auditing in which homomorphic linear authenticator with random masking is used for auditing process. The second scheme is of Threshold Cryptography in which Capability List is used to ensure the data access control. Our first scheme ensures that TPA will not gain any knowledge about the sensitive data during auditing process. Our second scheme is to guarantee that malicious users could not misuse the data stored on clouds.

1.1 OVERVIEW

The DES algorithm is used with RSA algorithm to make the overall algorithm very secure. In second layer of security we apply Data Encryption Standard Algorithm for encrypted that data which was the output of first stage after applying DES and RSA algorithm makes more secure for uploading of data in clouds. Following are the features of the DES algorithm which makes it very useful.

1.2 Domain Overview

The Blockchain is an encrypted, distributed database that records data, or in other words it is a digital ledger of any transactions, contracts - that needs to be independently recorded. One of the key features of Blockchain is that this digital ledger is accessible across several hundreds and thousands of computer and is not bound to be kept in a single place. Blockchain chain has already started disrupting the financial services sector, and it is this technology which underpins the digital currency- bit coin transaction. With Blockchain technology in financial sector, the

participants can interact directly and can make transactions across the internet without the interference of a third party. Such transactions through Blockchain will not share any personal information regarding the participants and it creates a transaction record by encrypting the identifying information. The most exciting feature of Blockchain is that it greatly reduces the possibilities of a data breach. In contrast with the traditional processes, in Blockchain there are multiple shared copies of the same data base which makes it challenging to wage a data breach attack or cyber- attack. With all the fraud resistant features, the block chain technology holds the potential to revolutionize various business sectors and make processes smarter, secure, transparent, and more efficient compared to the traditional business processes. You get a history of activity, not just a snapshot in time. When you look at a regular database, you're getting a snapshot of data that's up to date in that moment in time. Blockchain do this too, but they also maintain a record of all the information that existed before. It's a database with history, if you like. There's no one, central point of attack.

The fact that blockchain is a decentralised way of storing and accessing data makes the whole system incredibly secure – because, unlike a centralised database, there's no one single point of entry for hackers. This makes it particularly useful for recording transactions in a secure manner and no centralised control. Because the system of record is decentralised and replicated in its entirety in multiple places, there's no need for a central administrator and all the costs and infrastructure that comes with it.

1.2.1 Blockchain

Blockchain technology is revolutionizing a wide range of industries. Forbes' Bernard Marr highlights several Blockchain applications, including entertainment, such as music-streaming service spotify; the food industry, such as for supply-chain logistics; and healthcare, such as for storage and use of medical records. While the possibilities for block chain's applications may constantly be growing, however, the best known may still be bit coin. Imagine two friends living far away from each other, and one would like to transfer money to the other using blockchain technology. As mentioned, blockchain is a decentralized system of secure and trusted distributed

databases. It's a distributed ledger that records and shares the transaction details across many nodes (computers) that are part of the network. Every participant has the same copy of the ledger, and it's immutable—once a record or a transaction is registered, it cannot be modified. Blockchain was initially introduced to timestamp digital documents and prevent tampering of records. In simple terms, a chain of blocks that contain information is called a blockchain. When a transaction occurs, its related information is recorded into a block. A transaction initiated in one corner of the globe can get registered on the block, and then that block is verified (validated) by the miners and then added to the main blockchain. A block contains aggregated transactions that a miner has to validate, and for doing that, the miner gets rewarded.

1.2.2 Components of a Block

Previous Hash

The previous hash is the attribute that connects a block to its previous block. It consists of the hash value of the previous block.

Data

This consists of the sender's address, the receiver's address, and the transaction amount. There could be multiple transactions among multiple senders and receivers, so each block consists of any number of transactions, and each transaction will have a sender's address, a receiver's address, and a transactional nonce.

Nonce

Bit coin uses a proof-of-work algorithm, and to execute the algorithm, a random value is used to vary the output of the hash value; this is called the nonce. Proof of work is the process of transaction verification.

Hash

The hash is like a digital fingerprint. To get the hash for the current block, the process takes an input value (the previous hash, the data, and the nonce) and produces an output value of a fixed length. Bit coin uses the SHA-256 hashing algorithm to generate a 256-bit-length hash. It looks something like a hexadecimal value.

Public Distributed Ledger

To recap, a blockchain is a decentralized public distributed ledger that is used to record transactions across many computers. For example, user A transfers money to user B, user B transfers to C, and C transfers to B. A distributed ledger is a database that is shared among all the users who are part of the blockchain network. The transactions are accessed and verified by users of the bit coin network, thereby making it less prone to a cyber attack. Let's take an example in which bit coin users are transferring money: Bella is trying to transfer money to John, John is trying to transfer money to Elsa, and Elsa is trying to transfer money to Jack. So these are the three transactions to be initiated. If these transactions were happening on a central ledger, it could get corrupted, and there is the chance of data tampering. To solve this problem, a public distributed ledger plays a vital role: It ensures that each user who is part of the cycle has a copy of the transaction details. In our example, Bella, John, Elsa, and Jack all have the same ledger—the distributed ledger.

Encryption

Blockchain eliminates unauthorized access by using the cryptographic algorithm SHA-256 to ensure that the blocks are kept secure. Each user in the blockchain has his or her keys: a private one and a public one. The private key is known only to the sender; it is also used to confirm if the origin of the transaction is legitimate. The public key is also used to identify the user uniquely, but the sender shares it with every transaction. It floats on the blockchain network. Let's take a look at a typical transaction verification process. Suppose a sender wants to send a message. The sender will pass the message through the hash function and generate a hash value of the message. After the hash value has been created, it is passed through a signature algorithm, and with the private key, a digitally signed document is created.

The original message, the digitally signed document, and the public key are then transmitted to the receiver. At the receiver's end, the transaction message is passed through a hash function to get a hash value, and that hash value is compared with the hash value obtained bypassing the digital signature and public key through a verification function. The hash function creates a unique digital fingerprint of data. The message is passed through the hashing function, and it generates a hash value.

This hash value is called a digital print, and it has a unique property: Any hashing function is a one-way function; it cannot be reversed.

You cannot decode the original value from the hashed value.

A blockchain is a continuously growing collection of blocks that are chained together to form an open ledger by using cryptography as a key ingredient. Each block in the blockchain is given an identity to mark that block as unique, and this is achieved by using the hash functions that will generate a digest for that block. The collision resistance property of the cryptographic hash function, A Bit of Cryptography, ensures that it is infeasible to find two blocks that will result in the same hash value.

As a result, the hash function guarantees the uniqueness of the identity created for the block. When a new block is created, it will back reference the previous block using the digest of the previous block, thus linking that block to the blockchain. Modifying any of the blocks would change the identity of that block due to the new hash value. As a result, this would break the chain, as one of the block references will be invalid due to the newly generated hash value. Therefore, it's infeasible to modify a block such that it generates the same hash value as before. This is due to the pre-image resistance property of the cryptographic hash function, which ensures that the data of the blocks cannot be predicted even if we possess the hash value. This is why, once a chain of blocks is created, the integrity of the chain is ensured as each block references the previous block. The only way to modify the data of a block is by modifying all the subsequent blocks by updating its reference to the previous block.

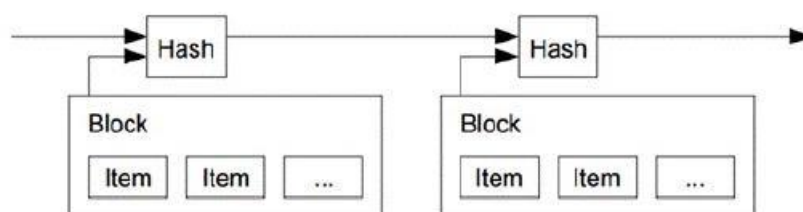


Fig 1.1: Hash diagram

Here Fig.1.1, represents the block linking process consists of several elements, such as creating a structure from the information, calculating the hash of the block, and appending it to the blockchain.

```
def __init__(self, index, previous_hash, timestamp, data, hash): self.index = index
```

```
self.previous_hash = previous_hash self.timestamp = timestamp self.data = data  
self.hash=hash
```

The preceding code snippet defines a Python class called Block that has all the basic attributes of a blockchain block. Usually, a block will contain both a header and a body, with the header containing metadata about the block. However, the preceding example doesn't distinguish Digital signatures are used in transactions due to their properties that ensure the integrity of the transaction contents and non-repudiation of any events in the transaction. A transaction embedded into the block will contain a certain action that is being signed by someone who possesses the private key. Owning the private key thus proves the identity of the signer.

The transaction can only be signed by the owner of the corresponding private key of the public key. In the case of wallet failure, seed information alone is sufficient to recover the private keys, so there is no need to back up all the keys in the wallet. The basic definition of a P2P network is a network where groups of independent computers called nodes are interconnected, sharing data without the assistance of any centralized servers. It is an architecture on top of the internet. Participants or nodes in this type of network are called peers because they are all equal and have equal responsibility within the network.

Since there are no special nodes in a P2P network, each peer is both a service provider and a consumer. P2P networking is an architecture where each peer acts as a server and a client simultaneously. Since a blockchain network is often implemented on a public network, it is difficult to create physical topology that is suitable for P2P networking.

To create this kind of architecture, a virtual or logical network overlay has to be constructed over the actual physical network topology. In Fig. 1.2, a logical network is created to achieve a convenient index of resources and peer discovery in a public network

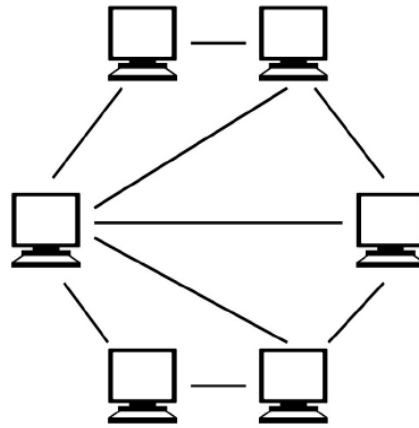


Fig 1.2: Network

There are two main classifications of a P2P network, based on how the nodes are linked:

unstructured and structured P2P networks. In an unstructured network, the peers aren't linked to each other in an organized way. Each node is randomly connected to the peers, forming a logical mesh. It is easy to build unstructured networks, and they are very robust due to the redundant distribution of nodes. However, these networks have drawbacks, such as the possibility of request flooding, an effect caused by a lack of knowledge about the distribution of resources. A structured P2P network overlay is formed by following a specific network topology to make sure that nodes can efficiently perform activities on the network. Creating a structured network ensures that a resource can be fetched from somewhere on the network in a certain time.

A distributed hash table (DHT) is a widely used structured network implementation that provides decentralized lookup service. Resource information in a DHT can be retrieved from hash tables using the key of a key/value pair stored in the table. The value associated with the key provides information about the peer that owns the resource. DHTs are also used in the Bit Torrent file sharing protocol as a substitute for centralized lookup services, such as trackers. Network discovery in a P2P network is crucial. No network is defined when a new node boots up. The new node must detect at least one blockchain node to be a part of the network. There are several ways in which a node can identify peers and thus discover a network. Different blockchain frameworks use their own protocols to perform peer discovery and efficient routing. We're going to start by exploring basic P2P network discovery by taking a look at Bit coin's original implementation.

Once a connection is set up between peers that have been discovered by the node, the node can query for information about other nodes that are connected to its peer. Similarly, the node can broadcast its own address information to the connected peers to improve its reachability. Each node also makes sure to maintain a threshold for the number of active connections in order to avoid unnecessary bandwidth usage. After the initial node handshake, packets are encapsulated as frames, which are then encrypted.

Each node that joins the blockchain network needs to update its local copy of the blockchain to synchronize its state with the global state of the rest of the network. This is achieved by block synchronization. A node that needs to update its blockchain sends a message consisting of blockchain height information. Any peer that has a longer blockchain sends an inventory consisting of metadata about the fixed number of blocks that needs to be added to the host node. Now the node makes a request to all its peers to fetch individual blocks by referring to the inventory it received. The node should make sure not to flood the network with block requests by maintaining a cap on the number of block requests it sends. Block synchronization is a long process for a newly-joined node. However, once all the blocks are up to date, it can verify the information in the block, such as the transactions on assets. The block synchronization process can be reinitialized whenever the node comes online after being inactive.

1.2.3 Proof of Work

Proof of work is a method to validate transactions in a blockchain network by solving a complex mathematical puzzle, and this process is called mining. Finding the nonce value is the mathematical puzzle that miners need to solve in the bit coin network, and it takes a huge amount of computational power and resources to find the nonce value. Users trying to solve the puzzle are called miners. The puzzle is solved by finding a nonce that generates a hash value and results in an output that is less than a predefined target. Miners verify transactions within a block and add the block to the blockchain when they have confirmed and verified the transaction. With proof of work, miners compete against one another to solve the mathematical puzzle; the first miner who solves the puzzle is rewarded and when a block is

resolved, the transactions contained in it are also considered valid, and the bit coins associated with the transactions then get deducted from the sender's account and move to the receiver's account. Although blocks are created by a miner node after validating all the previous blocks in the chain, it is the responsibility of each node in the network to perform a block validation to ensure that the block can be appended to the local copy of the blockchain.

Block validation is a simple process of checking whether the latest block has a pointer or hash reference pointing to the previous block. When a block includes complex data, such as a set of transactions, independent transactions have to be verified to validate the block. Since we will not be dealing with any complex data inside the block. Finding a global truth in the decentralized network is a strenuous task, and a variety of consensus algorithms help to achieve it. The proof-of-work algorithm, is the first and one of the best solutions. It ensures that the longest chain created has the most work contributed to it by legitimate nodes in the network. No consensus algorithm will be included in this application for the sake of simplicity. Although selecting a block may seem an obvious task, there will be situations when two or more nodes create a block at the same time that has different identities due to the different content included in its block. When a node receives different versions of a block of the same height, it will add one of them to the blockchain based on the order of arrival and either rejects the other or keeps it in a memory pool.

Due to this, nodes in the network might end up with different versions of the blockchain. This is called a temporary fork or a soft fork. Block exchange in this application is performed using both broadcasting query messages and blocks. A newly-joined node goes through an initial block synchronization process to update its local copy of the blockchain. When a new node joins a network, it connects to one of the available peers in the network; it then tries to exchange its block information with the peer, and updates the local blockchain, if required. The node sends a chain length or last block query message after connecting to the peer. Requesting the chain length is equivalent to requesting the last block to identify its index.

If the received block cannot be appended to the local blockchain due the hash mismatch, then the node sends a message to its peer asking it to send all the blocks. The node can also broadcast a message requesting all the blocks to ensure that it receives the longest chain. Nodes communicate with each other in a P2P network by

setting up multiple one-to-one connections, because there is no single peer that is completely trustable.

Maintaining multiple connections forces each node to broadcast any information to its connected peers in order to distribute data to the entire network and maintain the global truth.

The first one is Server, which will be dealing with the P2P communication on the blockchain network.

The other two classes are Blockchain and Block, which were covered in the previous chapter and perform basic blockchain operations. The Blockchain class also performs enhanced functions in order to validate the blocks received from the peers, as discussed in the application architecture

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY 1

Title	Third party public auditing on cloud storage using the cryptographic Algorithm
Authors	B. L. Adokshaja ; S. J. Saritha
Published Year	2017
Efficiency	✓ Secure and Highly efficient ✓ Less Encryption Time ✓ Support dynamic operation ✓ Privacy Preserving ✓ Batch Auditing
Drawbacks	↓ Less Secure, High Embedding extraction time. ↓ Poor Application Performance ↓ Reduce the Modality gap by synthesizing

Description	Data auditing mechanism is to be introduced in Cloud computing which deal with the secure data storage. Auditing is a process of checking the user data which can be done by the data owner or by a TPA. It helps to maintain the integrity of data stored on the cloud. The TPA work is divided into two: one is private auditability, which allows data owner and user can check the integrity of the data. No one has the authority to question the server regarding the data. But it tends to increases verification overhead of the user. Second is public auditability, only the TPA can check the confidentiality of the data. The TPA is an entity which is used so that it can act on behalf of the client. It has all the necessary expertise, capabilities, knowledge and professional skills which are required to handle the work of integrity verification and it also reduces the situation of the client. It is should be important that TPA should efficiently or frequently audit the cloud data storage without requesting for the local copy of data Cloud computing is worrying about many problems on the cloud like integrity and privacy. So it is required some more additional security and most efficient mechanisms which can get the integrity and privacy of data stored in the cloud. The proposed scheme consists of three basic entities; they are 1. Data owner, 2. Cloud server storage and 3. TPA. The data owner or the user is responsible for splitting the file into blocks, encrypting those blocks using AES algorithm, generating a hash value for each, to check the file corrupted or not. The cloud server is used only to store the encrypted blocks of files. Thus it has no additional burden of computing the verification proof. Here verification proof means generating of hashes for the encrypted blocks. This task is performed by
-------------	---

	TPA itself.
--	-------------

2.2 LITERATURE SURVEY 2

Title	Enhancing data storage security in Cloud using Certificate less public auditing
Authors	R. Swathi ; T. Subha
Published Year	2017
Efficiency	✓ Increases the opaqueness of the user data from the auditor ✓ Support scalable and efficient public auditing in the cloud computing. ✓ Provides a privacy preserving auditing process. ✓ Secure and Highly efficient
Drawbacks	↓ Narrowly specialized knowledge ↓ Heavyweight ↓ Big payloads
	Cloud computing is an innovative model to give beneficial, an on demand access to a typical configurable handling resources. In Cloud computing, IT-related abilities are given as administrations, open without requiring nitty gritty information of the fundamental advances, and with insignificant administration exertion. The considerable reserve Funds guaranteed by the cloud are however counter balanced by the apparent security dangers dreaded by users. The put away information at an untrustworthy server to confirm that the cloud server has first information exclusive of recovering it. The representation creates probabilistic ownership evidences through checking irregular arrangements of pieces from the cloud server, which has radically minimizes Input/output costs. The users' keeps up a consistent measure of data to check the data confirmation. The test/reaction convention sends a little, consistent information, which has minimises the organise correspondence.

	Sorting the data integrity into equal components permits for the detection of entity aspects that may make limitations with regards to the data integrity & the problems which are mapped to the significant measurement of integrity of data that they contact, recognize aspects that could menace the integrity of data for key data dependencies. To design and implement the efficient data storage in cloud the following goals are maintained and achieved. 1. Data security: Provides security to the data present in the cloud. Examples for data security include software/hardware disk encryption, backups. 2. Cloud storage: Cloud storage providers are in charge of maintain the data information an accessible, open and the physical environment secured & running. 3. Integrity: Maintaining end goal to guarantee the Trust worthiness and accessibility of information in cloud and uphold the nature of distributed storage benefit, so it is very weight to the users.
--	---

2.3 LITERATURE SURVEY 3

Title	S-Audit: Efficient Data Integrity Verification for Cloud Storage
Authors	Filipe Apolinário ; Miguel Pardal ; Miguel Correia
Published Year	2018
Efficiency	✓ Secure and Highly efficient ✓ Stateless verification ✓ Maintains Integrity of Data ✓ Maintains Confidentiality of Data
Drawbacks	□ Scalability is an issue for those designs as the number of users and the amount of data grow - Complexity of its Real Time Implementation - Prohibitively Expensive in terms of Time and Memory Usage

	This paper presents S-AUDIT, a service that provides integrity verification of data stored in commercial clouds. S-AUDIT uses homomorphic authentication with digital signatures to avoid retrieving the protected data from the cloud. The service was integrated with a cloud-backed file system called SCFS to show how it can be used in practice. Our experimental evaluation shows that using S-AUDIT is 7.1% cheaper than using RSA signatures when the integrity of the data is verified monthly, and 34.9% when verified weekly, in a typical setting. Nowadays data owners use integrity control mechanisms based on cryptographic hashes to protect their outsourced storage. Digital signatures are used for collaborative storage when data is shared among several cloud users, and MACs (Message Authentication Codes) are used for private storage when data is used by a single cloud user. For performing the integrity control, users need to have some personal key: an asymmetric private/public key pair for digital signatures or a symmetric key for MACs. The user data is stored together either with a signature or a MAC, computed with the user's private key or symmetric key, respectively. In S-AUDIT there is interaction among three types of entities: 1. clouds, 2. users and 3. auditors This paper propose the below cryptographic techniques: 1. Multiplicative Cyclic Group 2. Pairing-based cryptography 3. BLS Signature Scheme SW Scheme for Homo morphic Verifiable Integrity Proof
--	---

CHAPTER 3

AIM AND SCOPE

3.1 AIM OF THE PROJECT:

To improve the ability to collect and analyse data helps to improve effectiveness and quality of care.

3.2 SCOPE OF THE PROJECT:

Electronic health records possess the patient's medication details and their health history. The health records attract the attention of the attackers' as it possesses invaluable information. Loss of electronic health record leads to a wrong medication or surgery. Healthcare systems provide fewer security measures to secure the health records.

The main contributions of this project therefore are:

- Data Analysis
- Dataset Pre processing
- Training the Model
- Testing of Dataset

CHAPTER 4

METHODS AND ALGORITHMS USED

Hardware & Software Specification:

4.1 Hardware Specification:

Hardware Requirement

Hardware	Minimum Requirement
----------	---------------------

Disk Space	32 GB or more,10 GB or more for Foundation Edition
------------	--

Processor	1.4 GHz 64 bit
-----------	----------------

Memory	512 MB
--------	--------

Display	(800 × 600) Capable video adapter and monitor
---------	---

4.2 Software Specification:

Backend Technologies

✓ Python

✓ Numpy

✓ Sci-learn

✓ Eclipse IDE

Frontend Technologies

✓ Web Technologies

✓ Bootstrap

RAD Model

In Fig.4.1, Agile is an iterative, team-based approach to development. This method highlights the speedy delivery of an software application in comprehensive functional components

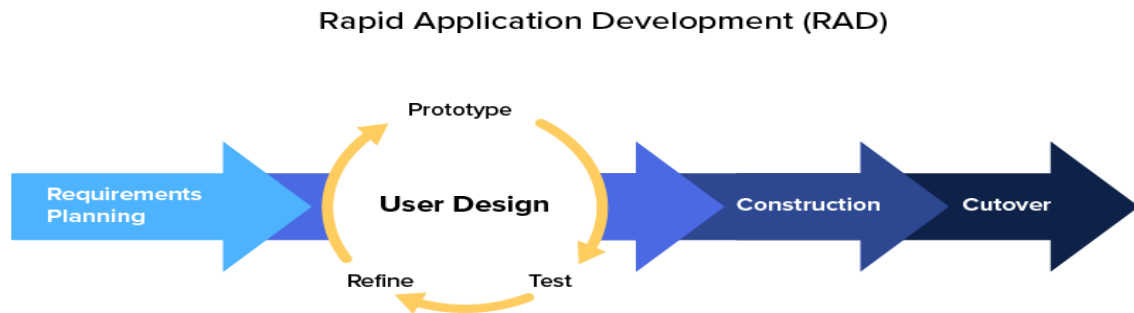


Fig 4.1: RAD model

Phase 1: Requirements planning

This phase is equivalent to a project scoping meeting. Although the planning phase is condensed compared to other project management methodologies, this is a critical step for the ultimate success of the project. During this stage, web application programmers, clients (software users), and team members communicate to determine the goals and expectations for the project as well as current and potential issues that would need to be addressed during the build. A basic breakdown of this stage involves:

Researching the current problem

Defining the requirements for the project

Finalizing the requirements with each stakeholder's approval

It is important that everyone has the opportunity to evaluate the goals and expectations for the project and weigh in. By getting approval from each key stakeholder and web application programmer, teams can avoid miscommunications and costly change orders down the road.

Phase 2: User design

Once the project is scoped out, it's time to jump right into development, building out the user design through various prototype iterations. This is the meat and potatoes of the RAD methodology—and what sets it apart from other project management strategies. During this phase, clients work hand in hand with web

application programmers to ensure their needs are being met at every step in the design process. It's almost like customizable software development where the users can test each prototype of the product, at each stage, to ensure it meets their expectations. This method gives web application programmers the opportunity to tweak the model as they go until they reach a satisfactory design. Both the software web application programmers and the clients learn from the experience to make sure there is no potential for something to slip through the cracks.

Phase 3: Rapid construction

Phase 3 takes the prototypes and beta systems from the design phase and converts them into the working model. Because the majority of the problems and changes were addressed during the thorough iterative design phase, web application programmers can construct the final working model more quickly than they could by following a traditional project management approach.

The phase breaks down into several smaller steps:

Preparation for rapid construction

Program and application development

Benefits of RAD methodology

RAD is one of the most successful software development programs available today, with numerous benefits for both software development teams as well as their clients. Here are just a few advantages:

- ☐ RAD lets you break the project down into smaller, more manageable tasks.
- ☐ The task-oriented structure allows project managers to optimize their team's efficiency by assigning tasks according to members' specialties and experience.
- ☐ Clients get a working product delivered in a shorter time frame.
- ☐ Regular communication and constant feedback between team members and stakeholders increases the efficiency of the design and build process.

4.1 Existing System

People store their data on cloud storage very commonly now a day. Security is a major issue in storing data on clouds. Cryptography techniques are very useful to impose security on data. In this paper a hybrid cryptography system is proposed to provide better security on the data which is stored on cloud storage. The proposed approach use RSA algorithm and DES algorithm and provide a hybrid of the two algorithms to provide more security on the data before storing it on cloud. The

proposed algorithm is implemented in JAVA and test on a sample plain text. The paper will be very useful for IOT applications storing data on cloud. It is verified that the proposed algorithm is working well to provide more security on data. The existing system uses the hybrid of the RSA and DES algorithms to store the data on the cloud storage. Before uploading data on cloud, the first step of security apply using RSA algorithm .where data will be encrypted by Rivest Shamir Adheman (RSA) Algorithm. Following are some advantages of the RSA algorithm which make it very useful:

The information which is encrypted and decrypted using RSA algorithm is only accessible to its sender and the receiver. No one in between can access the information. The RSA algorithm ensures that the content that is to be transferred is not altered during the exchange. The information exchanged via RSA algorithm is highly authenticated i.e. only the valid receiver is able to decrypt the information. The private key which is used by the RSA algorithm is unique. The DES algorithm is used with RSA algorithm to make the overall algorithm very secure. In second layer of security we apply Data Encryption Standard Algorithm for encrypted that data which was the output of first stage. After applying DES and RSA algorithm makes more secure for uploading of data in clouds. Following are the features of the DES algorithm which makes it very useful.

4.2 Drawbacks of Existing System

- ❖ In the cloud normally, files are stored as normal files. □ There is no way to check whether the files are corrupted or not. □ There is no proper verification methodology is available to check whether the data is stored correctly in the cloud or not.
- ❖ These data might be corrupted or lost due to the inevitable software bugs, hardware faults and human errors.
- ❖ There is no auditing mechanism to check whether the data is corrupted or not.

4.3 Proposed System

The proposed system uses two techniques first one is a public auditing and second one is threshold cryptography. Proposed system considers four key entities of cloud storage system such as Data Owner (DO) who can be an organization or an individual who wants to store his data on cloud storage; cloud storage system has large amount of storage space for the users data. Cloud Service Provider (CSP) maintain that storage space and is responsible for the security and availability of the owners data.

The Third Party Auditor (TPA), is third entity who checks the integrity if the sensitive data stored on cloud storage on side of the Data Owner and fourth one is cloud Data Users (DU) which uses data owners data stored on cloud. According to it the capability list is updated which stores information about user and their access rights to the data. Threshold cryptography scheme ensures security from mischievous users as while sending symmetric key Data Owner splits it into number of parts and every user gets the part of the decryption key. The novel third-party auditor scheme ensures data confidentiality using encryption techniques leading to secure authentication. Data Privacy by authenticating and secret sharing achieve mutual authentication between the client and the server. This method reduces the information cost in secret sharing.

The system architecture comprises of various physical entities that constitute the entire ecosystem. Here we would be talking about all different actors that constitute the Cloud, their roles, basic functionalities and the security services which our system provides. The system ensures proper authorization by only allowing system admin to have access to decryption keys. It is only the Cloud admin who is aware of the salted value added to every user password and Decryption Key before being saved in the database. Only a legitimate user will be allowed to access services and data being hosted on Cloud.

4.4 Advantages of Proposed System

- ✓ Data are stored with signature in the cloud.
- ✓ Remote Data Integrity Auditing.

- ✓ Data Integrity Electronic Health Records (EHR).
- ✓ Identity-Based Shared Data.
- ✓ provable Data Possession.
- ✓ Preserve Data Privacy.
- ✓ Sensitive Information Hiding

System Implementation:

4.5 List of Modules

Module 1 : Doctor Information

Module 2 : patient & EHR

Module 3 : Blinded File & Sanitized File Generation

4.5.1. Module 1 : Doctor Information

This system aim to maximize the capabilities of hospitals, multi-specialty clinics, doctors and medical practitioners by automating the processes of recording patient's information. Here common programming paradigm is procedural programming which structures a program like a recipe in that it provides a set of steps, in the form of functions and code blocks, which flow sequentially in order to complete a task. We use a standard name – to recognize immediately by the reader.

Login Screen & Dashboard

To access the Identity-Based Integrity Auditing application, you must first provide a username and password. Authentication means identifying a user and verifying that this user is allowed to access the Application. To access your Identity-Based Integrity Auditing Application, type the below URL in your web browser website address bar:

<http://127.0.0.1:9091>

Before being allowed to use the site application tools, you will be asked to login. The

login screen for the Identity-Based Integrity Auditing Application Admin asks for a username and password. Your username and password will be provided by your application user. Login information is case-sensitive. Once you have successfully logged in, you will be greeted by the Identity-Based Integrity Auditing Application Dashboard in the Admin area. The following goals are maintained and achieved.

Data security: Provides security to the data present in the cloud. Examples for data security include software/hardware disk encryption, backups.

Cloud storage: Cloud storage providers are in charge of maintain the data information an accessible, open and the physical environment secured & running.

Integrity: Maintaining end goal to guarantee the trustworthiness and accessibility of information in cloud and uphold the nature of distributed storage benefit, so it is very weight to the users. Electronic IoT information systems have the potential to improve health by giving health professionals improved information about their IoT. They can also improve the quality of health care and help control costs through improved efficiency. The literature review covers health information systems with an emphasis on the management of IoT information and will help contextualize the survey results. In particular it considers issues relating to the transfer of IoT information from paper-based records to digital.

4.5.2 Module 2- Patient &EHR:

This is the module where in all the patient demographic details can be taken care off. A permanent registration number, which will act as a patient identifier is allocated here. The registration number give to the patient will be storing the details of number of visits to the O.P.D. as well as number of admissions in the hospital.

Distributed health data networks-to improve the ability to collect and analyze data helps to improve effectiveness, safety and quality of care.

EHR RECORD:

EHRs have the ability to use multimedia including images which helps us to provide comprehensive patient information. Translation is one such example—healthcare providers and insurers are legally required to provide language access in the provision of services without compromising the security of sensitive patient data. Health care data encryption is a form of data security whereby electronic medical

records (EHR) are disguised so that unauthorized users may not read or make sense of them.

IoT Data

IoT Information

This is the module where in all the IoT demographic details can be taken care off. A permanent registration number, which will act as a IoT identifier is allocated here. The registration number give to the IoT will be storing the details. Individual IoT information is often aggregated for monitoring and evaluation functions or reporting to higher management levels. However, aggregated individual IoT information is also valuable in the local context where it can help to establish a 'culture of improvement', i.e. to measure quality of services and outcomes and use this to introduce changes. The purpose of collecting and storing IoT information is to make it available for decision making at a point of care or for analysis and action for management and policy. It is important to realize that most individual IoTs have more than one point of care. They may move from one location to another, including from country to country. It is essential that their individual IoT information be available at all points of care and all sites for analysis. In order for this to work, there must be standards for representing the data and for communication. Distributed IoT data networks have been proposed to improve the ability to collect and analyse data across institutions leading to improved effectiveness, safety, and quality of data.

4.5.3 Module 3 : Blinded File & Sanitized File Generation

Why Encryption is important?

- For maintain security of financial and other R&D data with various internal data which needs to be safe from intruders.
- The employees or the organizations do not want their financial or private items to be made public.
- To be make quick transitions so that private items are not made public. Encryption scrambles text to make it unreadable by anyone other than those with the keys to decode it, and it's becoming less of an added option and more of a must-have element in any security strategy for its ability to slow down and even deter hackers from stealing sensitive information. Healthcare organizations must encrypt their IoTs' data, they also need to know what they're encrypting and where in their

infrastructure. Encrypting data at rest is obviously different than encrypting data in motion. Algorithm for Blinded File:

- Python accepts the file input and encrypts it using Pycrypto module.
- The filename is taken as input parameter along with the password.
- Encryption is achieved with the help of key which is generated with algorithmic standards. The encrypted file is saved in the same directory with a prefix of(encrypted) added to it.
- Cryptographic hash function generates keys which helps in maintaining security of files which are used for symmetric encryption of files.
- The files are basically protected with passwords.
- Decryption follows reverse procedure where password and encrypted file is taken input parameters. Encryption algorithms transform their input data (called plaintext) in some way that is dependent on a variable key, producing cipher text; this transformation can easily be reversed, if (and, hopefully, only if) one knows the key. The key can be varied by the user or application, chosen from some very large space of possible keys. AES or Advanced Encryption Standards is one of the most widely used methods for encrypting and decrypting sensitive information. The key size used for this cipher specifies the number of repetitions or “rounds” required to put the plaintext through the cipher and convert it into cipher text.

Here’s how the cycles break down.

- 10 rounds are required for a 128-bit key
- 12 Rounds are required for a 192-bit key
- 14 Rounds are required for a 256-bit key

While longer keys provide the users with stronger encryptions, the strength comes at the cost of a reduction in performance, meaning that they will take longer to encrypt.

Advanced Encryption Standard is built from three block ciphers: AES-128, AES-192, and AES-256. Each of these encrypts and decrypts data in chunks of 128 bits by using cryptographic keys of 128-, 192- or 256-bits. The first step in the AES encryption process is substituting the information using a substitution table; the second transmutation changes data rows and the third shifts columns. The last transformation is a basic exclusive XOR process done on each column using a different part of the encryption key. The longer the encryption key, the more rounds are needed. Although hashing isn’t an encryption method, it is sometimes incorrectly referred to as one. Instead, hashes are a one-way function for providing

authentication. The hash function takes a larger file as input, processes it, and returns a smaller output that is almost guaranteed to be a unique “fingerprint” of the file. This makes it easy to compare two files to see if they are different from each other. Even changing one character will result in a different hash output. Hashing is often used in collaboration with encryption.

Third Party Auditor

Public verification, the client can resort to an independent third party auditor (TPA) to verify the storage correctness in the cloud on behalf of himself through public verification; Support dynamic operation, the outsourced data might not only be accessed but also updated by the users the TPA can concurrently handle multiple tasks from different clients at one time. Through these techniques, we can truly utilize a checking scheme to ensure the correctness of the data in the cloud, but cannot assure that the designed scheme is a high efficient. Considering various kinds of accessed devices, e.g., mobile devices, the client does not spend too much computation resources on both the initialization and verification phase for executing auditing protocol. Once after checking it then sends the report to the users'. To confirm the data reliability during the auditing process & the server generates the proof and randomly selects the blocks. The TPA then authenticates the proof against cloud server & the auditing result is sent to the user. A detailed execution analysis exhibited that the proposed approach gives preferred execution over the current works. The signed proof is sent to the trusted third party auditor (TTPA) for verification. The auditor first verifies the signature and for the validation of the proof. Auditor can anytime challenge the server for the integrity of the data. Then it checks and prove the response from the server is a valid one.

Our scheme consists of two important phases in the second stage.

A. Challenge response phase

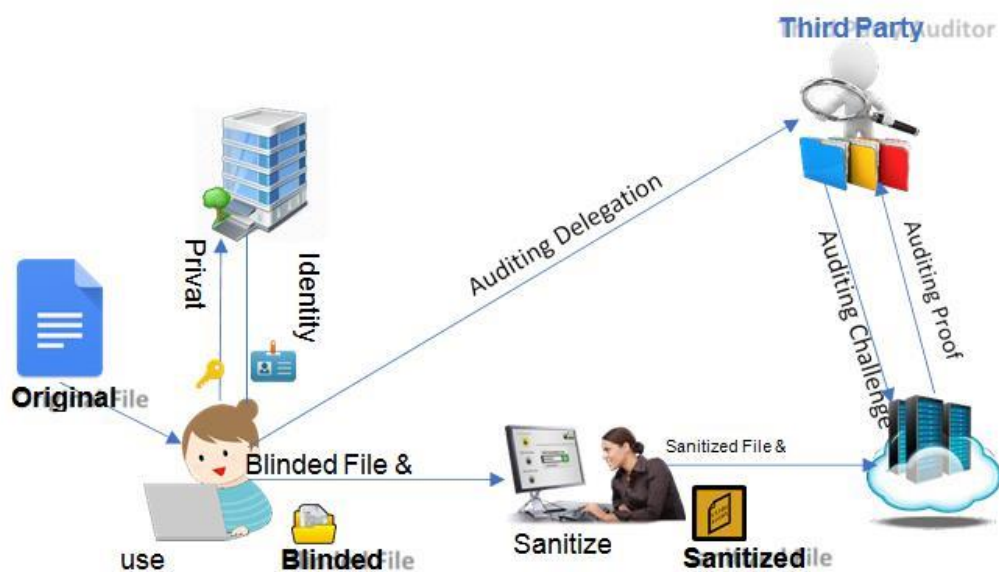
Step 1: Auditor sends a challenge request for the randomly selected blocks. Auditor also specifies the block positions to be checked by server. Challenge message contains the number of queried blocks. It randomly picks the element from the subset $S = \{ S_1 < \dots < S_c \}$ of set belongs to $[1, n]$. Auditor sends the challenge message to the cloud server. Chal msg= $\{(i, m_i)\} S_1 < i < S_c$

Step 2: Server takes an input file F , challenge request query and the Signature set S after receiving the challenge request generated by TTPA. He sends the proof as the response to the auditor.

B. Proof Verification phase

Step 3: Auditor (TTPA) checks for integrity after receiving the response by running the algorithm. As a next step, newly computed signature is compared with the previously stored signature as a first step, upon receiving the signature TTPA first checks the Certificate using its verification algorithm. If it is proved, then the response is verified using its public key (PK). Otherwise the integrity verification is not continued. As a second step, the proof response is further decrypted using public key (PK). This is done after the certificate is verified for checking authenticity. Then the signature is verified to prove the authentication of server. Our scheme is able to detect the corrupted data by an active adversary. Our auditing protocol uses digital signature algorithm in association with certificates. This certificate is used to check the authenticity of server. Certificates bind identity information like user name, ip address and social security number (SSN) to a public key with the help of digital signatures. Hence, no one is able to access the data.

Fig 4.2: ARCHITECTURE DIAGRAM



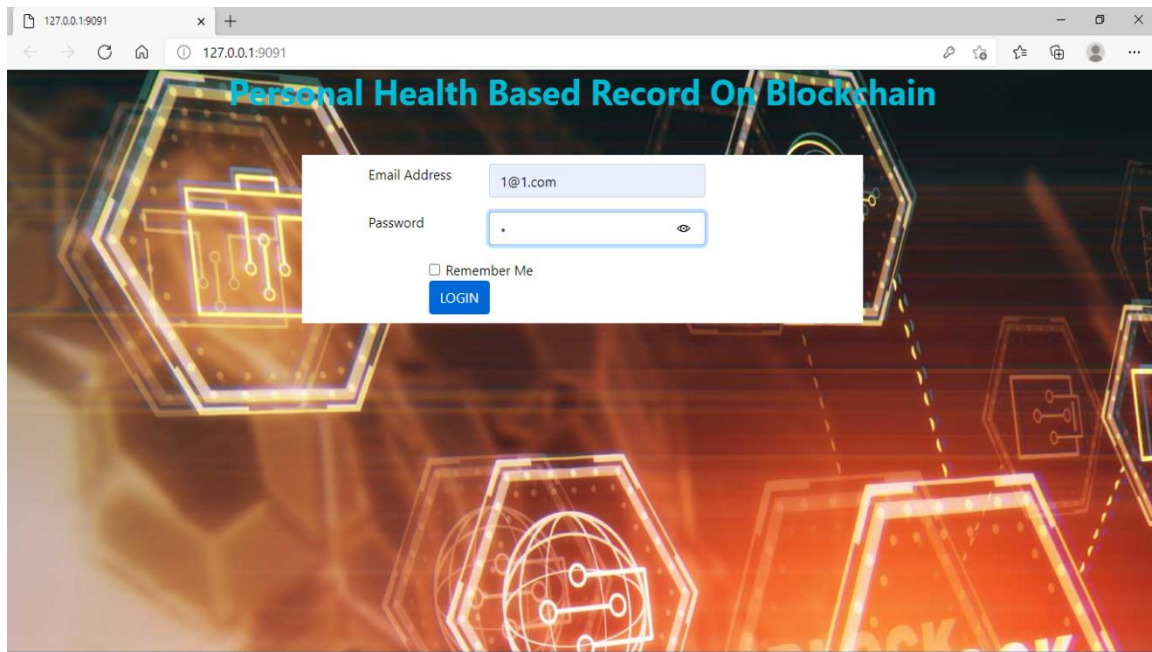
In Fig.4.2, an architecture diagram is a graphical representation of a set of concepts, that are part of an architecture, including their principles, elements and components. For system developers, they need system architecture diagrams to understand, clarify, and communicate ideas about the system structure and the user requirements that the system must support. It's a basic framework can be used at the system planning phase helping partners understand the architecture, discuss changes, and communicate intentions clearly.

CHAPTER - 5

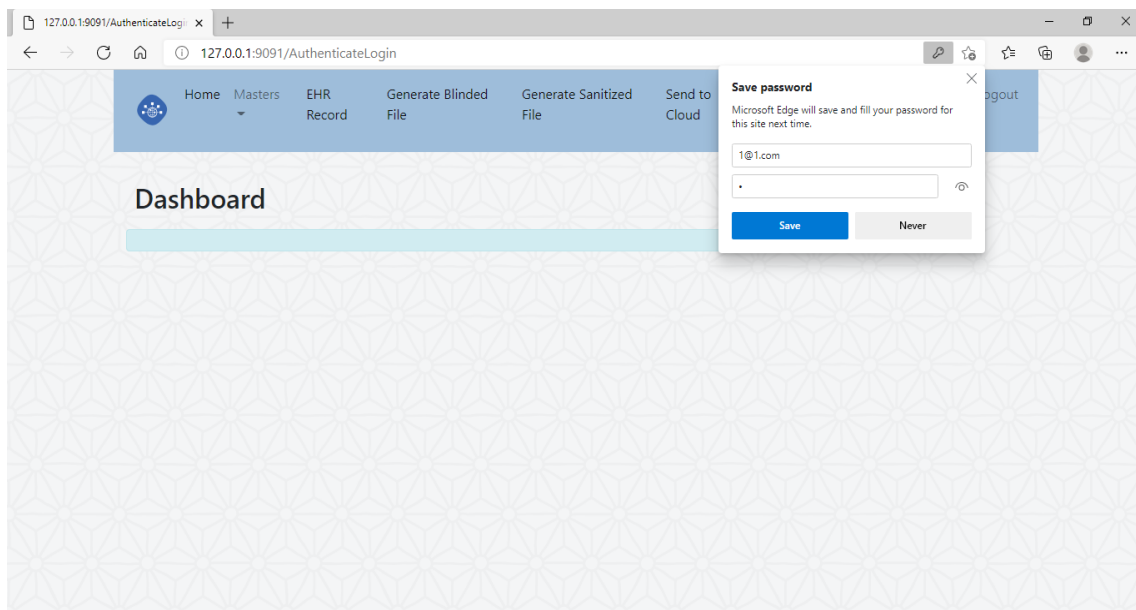
RESULTS AND DISCUSSION

SCREENSHOTS:

5.1 refers to, this is the main login page for personal health based record based on blockchain.



5.1 LOGIN PAGE



5.2 DASHBOARD

In 5.2, it shows the dashboard it includes EHR record, Blinded and sanitized files.

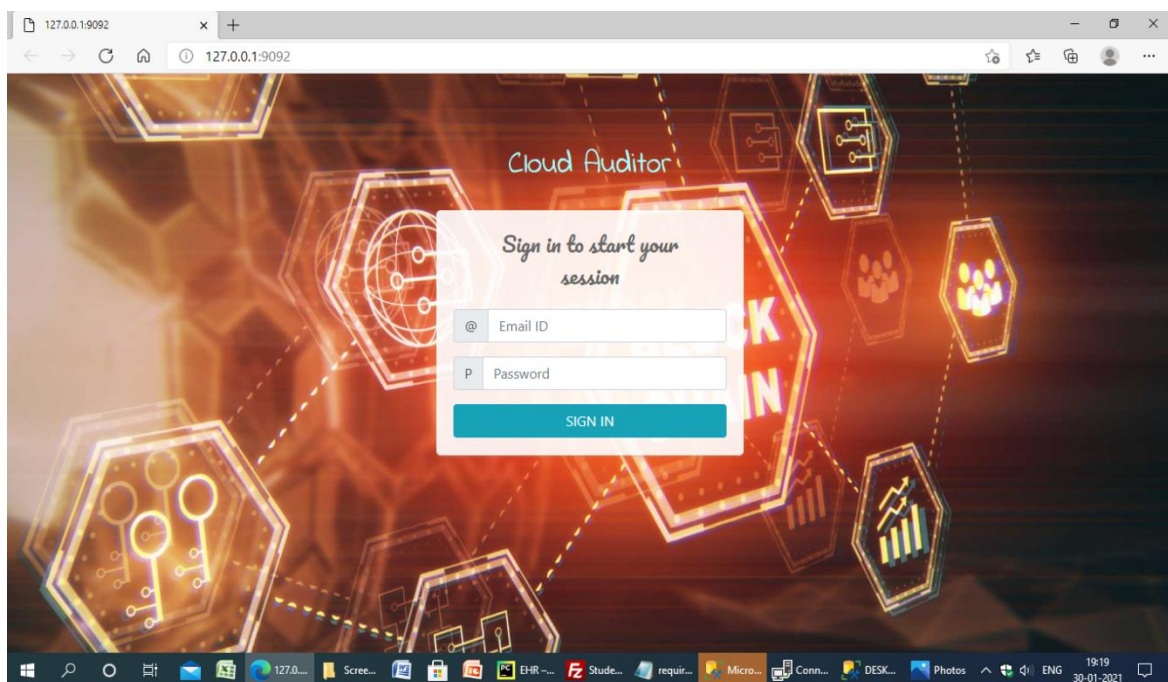
The screenshot shows a web browser window with the URL `127.0.0.1:9091/EHRRecordOperation?operation=Create`. The application has a navigation bar with links: Home, Masters, EHR Record, Generate Blinded File, Generate Sanitized File, Send to Cloud, Reports, Blockchain, System, and Logout. The main content area is titled 'Create EHR Record' and contains the following form fields:

- Eff Date:** A text input field containing '30/01/2021'.
- Doctor Name:** A dropdown menu with 'Dr. Sundar' selected.
- Patient Name:** A dropdown menu with 'sagari' selected.
- Disease:** A text input field containing 'pain'.
- Prescription To Upload:** A 'Choose File' button followed by the text 'prescription'.

At the bottom of the form are two buttons: 'Create' and 'Cancel'.

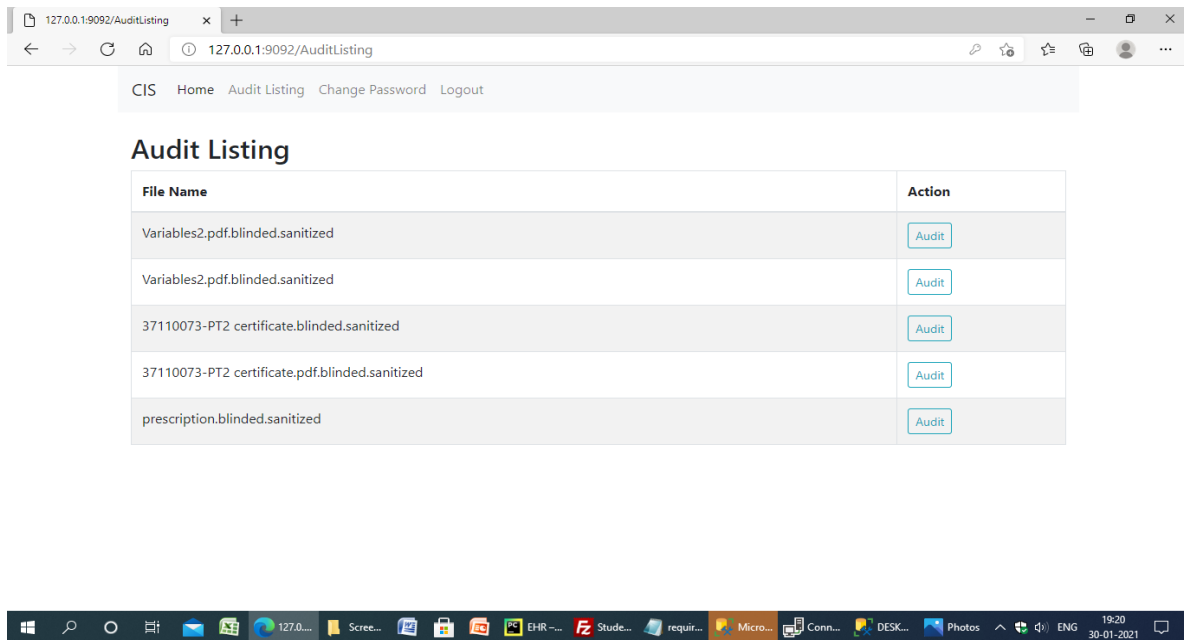
5.3 CREATE EHR RECORD

5.3 refer, after submitting the details EHR record will be created.



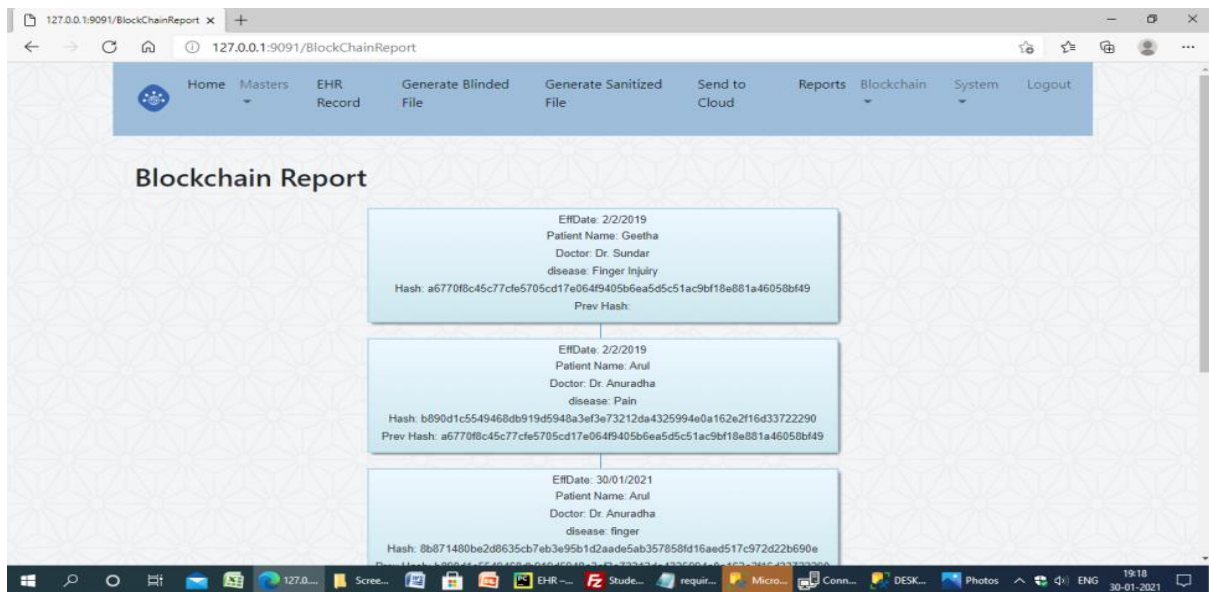
5.4 CLOUD AUDITOR

In 5.4 it represents cloud auditor, cloud auditor here admin can submit their details for sign in, only users can handle this auditor.



5.5 AUDIT LISTING

In 5.5 Audit Listing, The generated files will be visible and can audit those files.



5.6 BLOCKCHAIN REPORT

In 5.6, Blockchain report is created, in this report Effective date, Patient name, Doctor name, Disease, Hash number and previous hash number will be displayed.

CHAPTER 6

SUMMARY AND CONCLUSION

6.1 CONCLUSION:

The privacy and security of patient health information is a top priority and we used the block chain technology to keep information safe and secure. We use data encryption (form of data security) to prevent it from unauthorized users. Distributed health data networks –to improve the ability to collect and analyse data helps to improve effectiveness and quality of care.

REFERENCES

1. Awasthi and K. Singh, "A Key Agreement Algorithm Based on ECDSA for Wireless Sensor Network," in 3rd International Conference on Advanced Computing Networking and Informatics, 2015.
2. Arabat Rashmi Vinod, Bhalke Sumit Sunidatta, Kumari Uma Rani, and Pillai Preethy Sasidharan. Hindering Data Theft Attacks Through Fog Computing. International Journal of Research in Engineering and Technology 2014, Volume 3, pp. 427-429.
3. Dnyanesh Patil, Suyash Patil, Deepak Pote, and Nilesh Koli. Secured Cloud Computing With Decoy Documents. International Journal of Advances in Computer Science and Cloud Computing 2014, Volume 2, pp. 43-45.
4. H. liu, Y. huang, and J. K. Liu, "Secure sharing of Personal Health Records in cloud computing: Ciphertext-Policy Attribute-Based Signcryption," Future Gener. Comput. Syst., vol. 52, pp. 67–76, Nov. 2015.
5. J. Shao, R. Lu, and X. Lin, "Fine-grained data sharing in cloud computing for mobile devices," in Proc. IEEE Conf. Comput. Commun. (INFOCOM), Apr./May 2015, pp. 2677– 2685.
6. J. Zhou et al., "Securing outsourced data in the multi-authority cloud with fine-grained access control and efficient attribute revocation," Comput. J., vol. 60, no. 8, pp. 1210– 1222, Aug. 2017.

7. K. Huang et al., "PKE-AET: Public key encryption with authorized equality test," *Comput. J.*, vol. 58, no. 10, pp. 2686–2697, Oct. 2015.
8. L. Wu, Y. Zhang, K.-K. R. Choo, and D. He, "Efficient and secure identitybased encryption scheme with equality test in cloud computing," *Future Gener. Comput. Syst.*, vol. 73, pp. 22–31, Aug. 2017
9. M. Sriram, V. Patel, D. Harishma, and N Lakshmanan. A Hybrid Protocol to Secure the Cloud from Insider Threats. In *Cloud Computing in Emerging Markets (CCEM)*, IEEE International Conference, Bangalore, 2014.
10. R. Ahuja, S. K. Mohanty, and K. Sakurai, "A scalable attribute-set-based access control with both sharing and full-fledged delegation of access privileges in cloud computing," *Comput. Elect. Eng.*, vol. 57, pp. 241–256, Jan. 2017.
11. Sonali Khairnar and Dhanashree Borkar. Fog Computing: A New Concept to Minimize the Attacks and to Provide Security in Cloud Computing environment. *International Journal of Research in Engineering and Technology* 2014, Volume 3, pp. 124-127.
12. S. Jin-Shu, C. Dan, W. Xiao-Feng, and S. Yi-Pin, "Attributed-based encryption schemes," *J. Softw.*, vol. 22, no. 6, pp. 1299–1315, 2011.
13. Y. S. Rao, "A secure and efficient ciphertext-policy attribute-based signcryption for personal health records sharing in cloud computing," *Future Gener. Comput. Syst.*, vol. 67, pp. 133–151 Feb. 2017.
14. Z. Fu, X. Sun, S. Ji, and G. Xie, "Towards efficient content-aware search over encrypted outsourced data in cloud," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2016, pp. 1–9.
15. Zou, Y. Xiang, and G. Min, "Privacy preserving in cloud computing environment," *Secur. Commun. Netw.*, vol. 9, no. 15, pp. 2752–2753 Oct. 2016

APPENDIX

A. AUDIT SERVER

```
from flask import Flask, request, render_template, redirect, url_for
import os
import pypyodbc
import pyAesCrypt
from DoctorModel import DoctorModel
from PatientModel import PatientModel
from EHRRecordModel import EHRRecordModel
from UserModel import UserModel
import ftplib
import hashlib

app = Flask(__name__)
app.config["CACHE_TYPE"] = "null"

msgText = ""
msgType = ""

def initialize():
    global msgText, msgType
    msgText = ""
    msgType=""

@app.route("/")
def home():
    global msgText, msgType
    return render_template('Login.html')
```

```

@app.route("/AuthenticateLogin", methods=['POST'])
def authenticateLogin():
    global msgText, msgType
    emailid = request.form['emailid']
    password = request.form['password']

    conn1 = pypyodbc.connect('Driver={SQL Server};Server=DESKTOP-RO9FD6R\SQLEXPRESS;Integrated_Security=true;Database=IndentityBasedIntegrityAuditingV1', autocommit=True)
    cur1 = conn1.cursor()
    sqlcmd1 = "SELECT * FROM Users WHERE emailid = '"+emailid+"' AND password = '"+password+"'";
    print(sqlcmd1)
    cur1.execute(sqlcmd1)
    row = cur1.fetchone()
    cur1.commit()
    if not row:
        return render_template('Login.html')
    return render_template('Dashboard.html')

```

```

@app.route("/UserListing")

```

```

def UserListing():

```

B. CLOUD AUDIT SERVER

```

@app.route("/AuditOperation")
def AuditOperation():
    global msgText, msgType
    operation = request.args.get('operation')
    unqid = ""

```

```

row = EHRRecordModel(0, "", "", "")
conn3 = pypyodbc.connect('Driver={SQL Server};Server=DESKTOP-
RO9FD6R\SQLEXPRESS;Integrated_Security=true;Database=IndentityBasedIntegri
tyAuditingV1', autocommit=True)
cursor3 = conn3.cursor()

if operation != "Create" :
    unqid = request.args.get('unqid').strip()
    conn2 = pypyodbc.connect('Driver={SQL Server};Server=DESKTOP-
RO9FD6R\SQLEXPRESS;Integrated_Security=true;Database=IndentityBasedIntegri
tyAuditingV1', autocommit=True)
    cursor = conn2.cursor()
    sqlcmd1 = "SELECT ehrRecordID, effDate, fileHashValue,
prescriptionFileName FROM EHRRecords WHERE ehrRecordID = '"+unqid+"'"
    cursor.execute(sqlcmd1)
    dbrow = cursor.fetchone()
    if dbrow:
        ftp = ftplib.FTP('101.99.74.37', 'Student1', '1Osk52#k')
        fileName = dbrow[3]
        print("fileName", fileName)
        ftp.retrbinary('RETR '+fileName+".blinded.sanitized",
open('static/DOWNLOADED_FILES/'+fileName+".blinded.sanitized", 'wb').write,
1024*1024)
        print("FTP DONE1")# send the file # close file and FTP
        ftp.quit()
        print("FTP DONE2")
        password = "K9I@#CDW"
        bufferSize = 64 * 1024
        try:
            with open(os.path.join('static/DOWNLOADED_FILES',
fileName+".blinded.sanitized"), "rb") as fln:
                with open(os.path.join('static/DOWNLOADED_FILES',
fileName+".blinded"), "wb") as fOut:
                    encFileSize = stat(os.path.join('static/DOWNLOADED_FILES',
fileName+".blinded.sanitized")).st_size
                    print("encFileSize", encFileSize)
                    pyAesCrypt.decryptStream(fln, fOut, password, bufferSize,
encFileSize)
                    hasher = hashlib.md5()
                    with open(os.path.join('static/DOWNLOADED_FILES',
fileName+".blinded"), "rb") as afile:
                        buf = afile.read()
                        hasher.update(buf)
                        currentHashValue = hasher.hexdigest()
                        print("currentHashValue", currentHashValue)
                        print(dbrow[2], currentHashValue)
                        isProblem = False
                        if dbrow[2] != currentHashValue:
                            isProblem = True
                        print(dbrow[2], currentHashValue)

```

```

        row = EHRRecordModel(dbrow[0], dbrow[1], fileHashValue=dbrow[2],
prescriptionFileName=dbrow[3] )

        return render_template('AuditOperation.html', row = row,
operation=operation, currentHashValue=currentHashValue,isProblem=isProblem)
    except ValueError:
        row = EHRRecordModel(dbrow[0], dbrow[1], fileHashValue=dbrow[2],
prescriptionFileName=dbrow[3] )
        return render_template('AuditOperation.html', row = row,
operation=operation, currentHashValue="",isProblem=True)
        return redirect(url_for("AuditListing"))

@app.route("/ProcessAuditOperation",methods = ['POST'])
def ProcessAuditOperation():

    unqid = request.form['unqid'].strip()
    isProblem = request.form['isProblem'].strip()
    print()
    if isProblem == "True":
        isProblem = 1
    else:
        isProblem = 0
    conn1 = pypyodbc.connect('Driver={SQL Server};Server=DESKTOP-
RO9FD6R\SQLEXPRESS;Integrated_Security=true;Database=IndentityBasedIntegri
tyAuditingV1', autoccommit=True)
    cur1 = conn1.cursor()
    sqlcmd = "UPDATE EHRRecords SET isAudited = 1, isProblem =
""+str(isProblem)+" WHERE ehrRecordID = ""+unqid+""

    cur1.execute(sqlcmd)
    cur1.commit()
    conn1.close()
    #return render_template('EHRRecordListing.html', processResult="Success!!!.
Data Uploaded. ")
    return redirect(url_for("AuditListing"))

if __name__ == "__main__":
    app.run(port=9092)

```