SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

A
MINI PROJECT REPORT
ON
“Medical Record”

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

IN THE FULFILLMENT OF THE REQUIREMENT

OF

Laboratory Practice IV

Fourth Year Computer Engineering

Academic Year 2024-25

BY

Name of Students:	**Roll No.:**
Pratik Abhang	4101001
Darshana Borase	4101030
Omkar Darekar	4101032
Chaitanya Gunde	4101056

Under the Guidance of

Mr. S. A. Shinde

DEPARTMENT OF COMPUTER ENGINEERING
STES’S SINHGAD INSTITUTE OF TECHNOLOGY AND SCIENCE
NARHE, PUNE – 411041

CERTIFICATE

This is to certify that,

Name of Students:	Roll No.:
Pratik Abhang	4101001
Darshana Borase	4101030
Omkar Darekar	4101032
Chaitanya Gunde	4101056

studying in SEM-VII, B.E. Computer Engineering, they have successfully completed their LP-III Lab Mini-Project report titled **“Medical Rcord”** under the supervision of **Mr. S. A. Shinde**, in fulfillment of the requirements for Laboratory Practice-III for the academic year 2024-2025, as prescribed by Savitribai Phule Pune University, Pune.

Mr. S. A. Shinde
Guide
Department of Computer Engineering

Dr. G. S. Navale
HOD
Department of Computer Engineering

Dr. S. D. Markande
Principal

SINHGAD INSTITUTE OF TECHNOLOGY AND SCIENCE, NARHE, PUNE– 411041

Place:

Date:

ACKNOWLEDGEMENT

We take this opportunity to acknowledge each and every one who contributed towards our work. We express our sincere gratitude towards guide **Mr. S. A. Shinde**, Assistant Professor at Sinhgad Institute of Technology and Science, Narhe, Pune for her valuable inputs, guidance and support throughout the course.

We wish to express our thanks to **Dr. G. S. Navale,** Head of Computer Engineering Department, Sinhgad Institute of Technology and Science, Narhe for giving us all the help and important suggestions all over the Work.

We thank all the teaching staff members, for their indispensable support and priceless suggestions. We also thank our friends and family for their help in collecting data, without their help **LP-III** report have not been completed. At the end our special thanks to **Dr. S. D. Markande,** Principal Sinhgad Institute of Technology and Science, Narhe for providing ambience in the college, which motivate us to work.

Name of Students:	**Sign:**
Pratik Abhang	
Darshana Borase	
Omkar Darekar	
Chaitanya Gunde	

CONTENTS

1. INTRODUCTION

A blockchain is a publicly-managed and verified record of transactional data. All of the data blocks are ordered chronologically and are connected to form a “chain” — hence, the term “blockchain.” All of the chain’s old blocks of data are permanent; they can’t be modified or altered retroactively.

Blockchain networks depend on a healthy population of full nodes, i. e. individual peers that store, verify and distribute the full set of data comprising the blockchain—the history of all past transactions. And blockchains are commonly advertised as being immutable, making it impossible to change or erase already posted data. The idea we are submitting uses the method called encryption and hashing to store the data in blockchain in an encrypted form so that no one without the key should be able to see the content of blockchain unless the permission is granted to the user.

This can be achieved if we effectively store the key in such a way that nobody can access it and when required it will be readily available. To achieve such a system we had to do some of the work off-chain i.e outside the blockchain. We created a third party system, this system is centralised system making it a perfect candidate to ensure privacy. The main objective of the third party system is to store a unique key of the data (patient data) in such a way that even this third party system will not be able to access the data even with the key stored in it.

Problem Statement:

Develop a Blockchain based application for health related medical records.

2. OBJECTIVE

- **Ensure Data Integrity**: Use blockchain's immutability to maintain a trusted and tamper-proof medical record system.
- **Improve Data Access**: Allow healthcare providers secure, real-time access to patient records with patient consent.
- **Enhance Data Ownership**: Empower patients by giving them full control over their personal health data.
- **Facilitate Auditing**: Provide a transparent, time-stamped audit trail of all access and changes to medical records.
- **Increase Interoperability**: Create a system that seamlessly works across different healthcare providers and institutions.
- **Automate Data Sharing**: Use smart contracts to automate the sharing of medical records under predefined conditions.
- **Ensure Regulatory Compliance**: Build a system that adheres to healthcare laws like HIPAA or GDPR to ensure patient data privacy.
- **Reduce Data Breaches**: Minimize the risk of data breaches by decentralizing and encrypting sensitive health information.

3. HARDWARE AND SOFTWARE REQUIREMENT

Hardware:

A minimum of 4GB RAM and multi-core CPU for development and running blockchain nodes.

Software:

- Ganache
- Solidity (for Ethereum) or Chaincode (for Hyperledger).
- VS Code
- React
- Node.js
- Metamask

4. ARCHITECTURE & COMPONENTS

- **Blockchain**

  A blockchain is a growing list of records, called blocks, that are linked together using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data (generally represented as a Merkle tree). The timestamp proves that the transaction data existed when the block was published in order to get into its hash.

- **Smart Contracts**

  A smart contract is a computer program or a transactional Protocol which is intended to automatically execute, control or document legally relevant events and actions according to the terms of a contract or an agreement. The objectives of smart contracts are the reduction of need in trusted intermediators, arbitrations and enforcement costs, fraud losses, as well as the reduction of malicious and accidental exceptions.

- **Ethereum**

  Ethereum is a decentralized, open-source blockchain with smart-contract functionality. Ether. (ETH) is the native cryptocurrency of the platform. After Bitcoin, it is the second-largest cryptocurrency.

- **Third Party System**

  This is an off-chain centralised system or server where all the keys can be stored. The Data is encrypted with the help of an encryption algorithm called AES (Advanced Encryption Standard) and the key is generated using an cryptographic hash function called SHA-256 (Secure Hash Algorithm 2).

- **Advanced Encryption Standard**

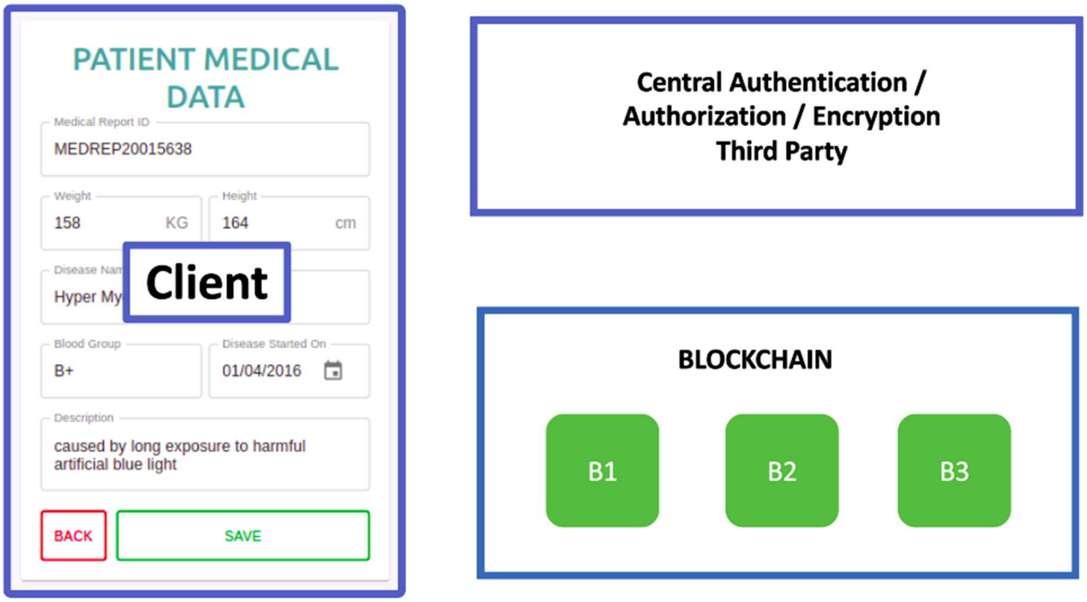
The AES algorithm (also known as the Rijndael algorithm) is a symmetrical block cipher algorithm that takes plain text in blocks of 128 bits and converts them to ciphertext using keys of 128, 192, and 256 bits. Since the AES algorithm is considered secure, it is in the worldwide standard.

- **SHA-256**

A cryptographic hash (sometimes called ‘digest’) is a kind of ‘signature’ for a text or a data file. SHA-256 generates an almost-unique 256-bit (32-byte) signature for a text.

- **Client**

The client can store the data (patient data) using front end technology (ex web app). Before sending the data to either the blockchain or third party system the data is encrypted locally using AES and SHA-256 is used to generate a key.



PATIENT MEDICAL DATA

Medical Report ID: MEDREP20015638

Weight: 158 KG

Height: 164 cm

Disease Nam: Hyper My

Blood Group: B+

Disease Started On: 01/04/2016

Description: caused by long exposure to harmful artificial blue light

BACK
SAVE
Client
Central Authentication / Authorization / Encryption Third Party
BLOCKCHAIN
B1
B2
B3

5. WORKING

Three components take part in the encryption, hashing and storage process and they are Client, Blockchain and the Third party system.

1. To store the encrypted data in the blockchain we follow the steps given below

The client first creates a key using the data as key-text for the SHA-256 cryptographic hashing function. Now the generated key is used to encrypt the data locally using the AES encryption algorithm.

Now the encrypted data is sent to the Third Party System where the encrypted data is again encrypted using the steps followed in step-1 i.e the encrypted data is used as key-text for SHA-256 which generates a key and this generated key is used to encrypt the locally encrypted data. The key generated is stored in the database of the server with the id of the owner of the data.

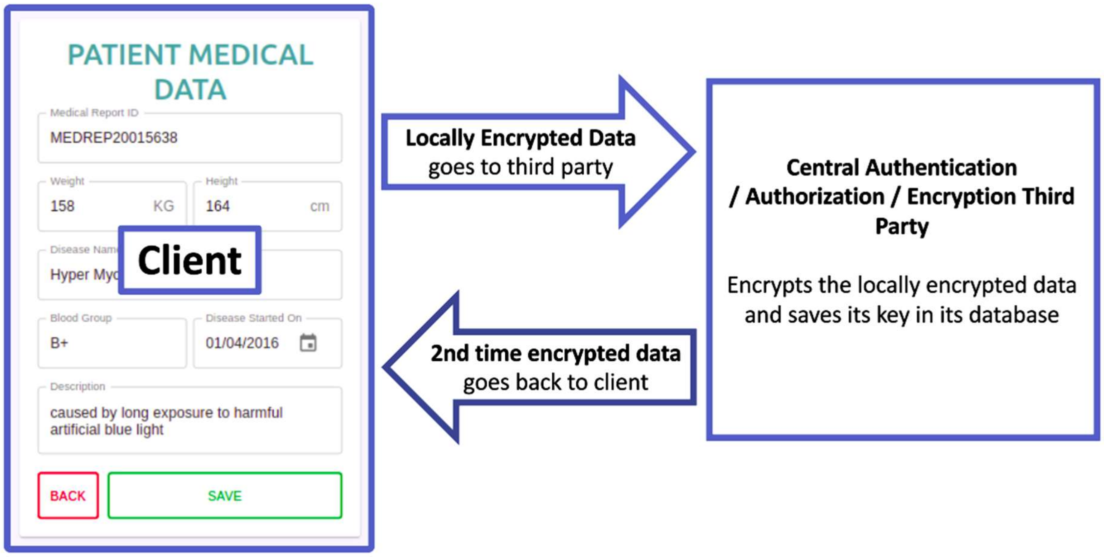
Now the encrypted data from the Third Party System is sent back to Client which sends this encrypted data from the Third Party System along with the locally generated key to the blockchain. The blockchain then writes this transaction in the block and stores the data in its database. Which means that every node connected to this blockchain network has this new encrypted data which is immutable.

2. To fetch encrypted data from the blockchain we follow the steps given below

The encrypted data and the locally generated key stored in the blockchain is fetched.

The encrypted data is then sent to Third Party System where it decrypts the data using the key stored in it’s database with user ID and we get the locally encrypted data but only if there is a permission by the owner of the data to share it with the owner of the request sent from the client. This decrypted data is then sent to the client.

The decrypted data from the Third Party System is then decrypted to original data using the key fetched from the blockchain. And we get the original data to show it to the world.



PATIENT MEDICAL DATA

Medical Report ID: MEDREP20015638

Weight: 158 KG

Height: 164 cm

Disease Nam: Hyper Myc

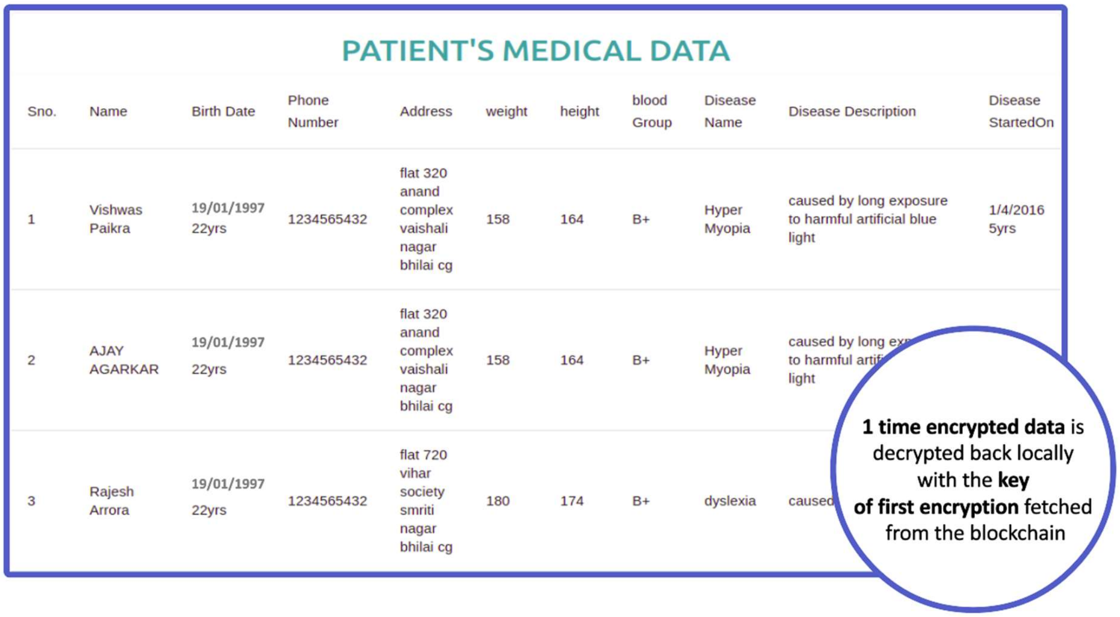
Blood Group: B+

Disease Started On: 01/04/2016

Description: caused by long exposure to harmful artificial blue light

BACK
SAVE
Client
Locally Encrypted Data goes to third party
Central Authentication / Authorization / Encryption Third Party
Encrypts the locally encrypted data and saves its key in its database
2nd time encrypted data goes back to client

6. RESULTS



PATIENT'S MEDICAL DATA

Sno.	Name	Birth Date	Phone Number	Address	weight	height	blood Group	Disease Name	Disease Description	Disease StartedOn
1	Vishwas Paikra	19/01/1997 22yrs	1234565432	flat 320 anand complex vaishali nagar bhilai cg	158	164	B+	Hyper Myopia	caused by long exposure to harmful artificial blue light	1/4/2016 5yrs
2	AJAY AGARKAR	19/01/1997 22yrs	1234565432	flat 320 anand complex vaishali nagar bhilai cg	158	164	B+	Hyper Myopia	caused by long exp to harmful artifi light	
3	Rajesh Arrora	19/01/1997 22yrs	1234565432	flat 720 vihar society smriti nagar bhilai cg	180	174	B+	dyslexia	caused	

1 time encrypted data is decrypted back locally with the **key of first encryption** fetched from the blockchain

7. CONCLUSION

We know that the data in blockchain is immutable i.e once the data is written in the blockchain it cannot be deleted and any body have access to the blockchain network can access the any data present inside the blockchain which is a major issue with blockchain especially if we want to create a database in the blockchain which needs to be deleted once its work is over or if any user wants its data to be removed from the database or if you don't want to share your data with everyone or you want to share your data with the selected few.

So in order to achieve a little privacy inside the blockchain we created this method where the original data is encrypted twice first locally and then in the Third party system. And when we send the data to the blockchain the data is in encrypted form so it means that if somebody tries to look inside the blockchain to steal the data for fault reasons it will be useless as the person will not be able to understand this data. In case if a user would one day wish its data to be deleted from the blockchain it can be apparently done by deleting all the keys associated with the data in order to decrypt the data.

It means that the key of the encrypted data present in the Third Party System’s database associated with its respective owner will simply be deleted. Once the key is deleted the encrypted data is nothing more than random bits which are totally useless and cannot be utilized in any shape or form. This is how our method works to achieve privacy within blockchain.

8. REFERENCES

1) “Blockchain” https://en.wikipedia.org/wiki/Blockchain
2) “Ethereum” https://ethereum.org/en/
3) “Smart Contracts” https://ethereum.org/en/developers/docs/smart-contracts/
4) “SHA-256” https://en.wikipedia.org/wiki/SHA-2
5) “AES” https://www.educative.io/edpresso/what-is-the-aes-algorithm
6) "How to Build Ethereum Dapp with React.js · Complete Step-By-Step Guide" https://www.dappuniversity.com/articles/ethereum-dapp-react-tutorial