A

Project Report

on

DYNAMIC HEALTH LEDGER

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

COMPUTER SCIENCE AND ENGINEERING

by

Anumula VarunChand
(20EG105302)

Pathlavath Pavan Raj
(20EG105339)

Shivaratri Aishwarya
(20EG105345)

Under the guidance of

Dr. T Shyam Prasad MTech., Ph.D

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ANURAG UNIVERSITY
Venkatapur (V), Ghatkesar (M), Medchal(D), T.S-500088
(2020-2024)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**DYNAMIC HEALTH LEDGER**" being submitted by **Anumula VarunChand** bearing the Hall Ticket number **20EG105302, Pathlavath Pavan Raj** bearing the Hall Ticket number **20EG105339** and **Shivaratri Aishwarya** bearing the Hall Ticket number **20EG10545** in partial fulfillment of the requirements for the award of the degree of the **Bachelor of Technology** in **Computer Science and Engineering** in **Anurag University** is a record of bonafide work carried out by them under my guidance and supervision from academic year 2023 to 2024.

The results presented in this project have been verified and found to be satisfactory. The results embodied in this project report have not been submitted to any other University for the award of any other degree or diploma.

Signature of Supervisor	**Signature of Dean**
Dr. T Shyam Prasad	Dr. G. Vishnu Murthy
M. Tech., Ph.D	M.Tech., Ph.D
Assistant Professor	Professor, CSE

External Examiner

DECLARATION

We hereby declare that the report entitled **"DYNAMIC HEALTH LEDGER"** submitted to the **Anurag University** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology (B. Tech)** in Computer Science and Engineering is a record of an original work done by us under the guidance of **Dr. T. Shyam Prasad, Assistant Professor** and this report has not been submitted to any other university for the award of any other degree or diploma.

Anumula Varun Chand
20EG105302

Pathlavath Pavan Raj
20EG105339

Shivaratri Aishwarya
20EG105345

Place: Anurag University,Hyderabad
Date:

ACKNOWLEDGEMENT

We would like to express our sincere thanks and deep sense of gratitude to project supervisor **Dr. T Shyam Prasad**, **Assistant Professor, Department of Computer Science and Engineering**, Anurag University for his constant encouragement and inspiring guidance without which this project could not have been completed. His critical reviews and constructive comments improved my grasp of the subject and steered to the fruitful completion of the work. His patience, guidance and encouragement made this project possible.

We would like to express special thanks to **Dr. V. Vijaya Kumar, Dean School of Engineering,** Anurag University, for his encouragement and timely support in our B. Tech program.

We would like to acknowledge my sincere gratitude for the support extended by **Dr. G. Vishnu Murthy**, **Dean, Department of Computer Science and Engineering,** Anurag University.

We also express my deep sense of gratitude to **Dr. V. V. S. S. S. Balaram**, Academic coordinator. **Dr. T. Shyam Prasad,** Assistant Professor**,** Project Coordinator and Project review committee members, whose research expertise and commitment to the highest standards continuously motivated us during the crucial stage of my project work.

Anumula VarunChand
20EG105302

Pathlavath Pavan Raj
20EG105339

Shivaratri Aishwarya
20EG105345

ABSTRACT

The development of the Dynamic Health Ledger marks a significant advancement in overcoming the complexities surrounding the security, sharing, and maintenance of health records within a distributed and decentralized healthcare framework. By harnessing the power of RSA for key exchange and seamlessly integrating blockchain technology, this innovative system ensures the integrity and confidentiality of sensitive health data. In response to the critical need for robust mechanisms to facilitate the secure sharing and management of health records, the Dynamic Health Ledger protocol has been meticulously crafted. Leveraging the RSA (Rivest-Shamir-Adleman) algorithm for secure key exchange provides a sturdy foundation for data encryption and access control, granting authorized stakeholders secure access to health records while upholding utmost confidentiality and privacy. Furthermore, the incorporation of blockchain technology elevates the protocol's resilience and transparency. Acting as a distributed ledger, the blockchain meticulously records all transactions and accesses pertaining to health records, establishing an immutable audit trail essential for regulatory compliance, accountability, and fostering trust within the healthcare ecosystem. Central to the development process of the Dynamic Health Ledger protocol is its performance optimization. Through strategic enhancements in data storage, retrieval, and sharing processes, the system achieves unprecedented levels of efficiency and scalability. Moreover, this project lays a robust foundation for future innovations, including the integration of machine learning algorithms to further enhance data analytics capabilities for healthcare providers.

Keywords – Distributed health records, Secure key exchange, Data encryption, Immutable, RSA algorithm, Blockchain technology.

TABLE OF CONTENTS

S. No.	CONTENT	Page No.
1.	Introduction	1
	1.1. Motivation	2
	1.2. Problem Definition	3
	1.3. Problem Illustration	3
	1.4. Objective	7
2.	Literature Survey	8
3.	Proposed Method	9
	3.1 Analysis	10
	3.1.1 RSA Key Exchange	10
	3.1.2 Blockchain Integration	10
	3.1.3 Scalability and User Experience	11
	3.2. RSA Implementation	12
	3.2.1 Key Generation	12
	3.2.1.1 Modulus and Euler's Totient Function	13
	3.2.1.2 Public and Private Exponents	13
	3.2.2 Encryption	13
	3.2.2.1 Plaintext Conversion	13
	3.2.2.2 Exponential Encryption	13
	3.2.2.3 Security Considerations	13
	3.2.3 Decryption	13
	3.2.3.1 Exponential Decryption	13
	3.2.3.2 Plaintext Reconstruction	14
	3.2.4 Security Considerations	14
	3.3. Blockchain Technology	15
	3.3.1 Decentralized Network	15
	3.3.2 Blocks	15
	3.3.3 Permissioned vs Permissionless Blockchains	16

4. Implementation 17

4.1. Functionality 17

4.1.1 Patient-Controlled Access 17

4.1.2 Interoperable Health Data Exchange 17

4.1.3 Immutable Audit Trail 17

4.2. System Architecture and Methodology 18

4.2.1 User Interface Layer 18

4.2.2 Application Layer 18

4.2.3 Blockchain Layer 19

4.2.4 Security Layer 19

4.3. RSA Key Generation 20

4.3.1 Encryption with Public Key 21

4.3.2 Decryption with Private Key 21

4.3.3 Secure Communication 21

4.4. Blockchain Implementation 21

4.4.1 Smart Contract Development 21

4.4.2 Data Verification 21

4.4.3 Access Control 22

4.4.4 Sample Code 22

4.4.4.1 main.py 22

4.4.4.2 RSA.py 23

4.5. Experiment Screenshots 26

4.5.1 login page for patient 26

4.5.2 Uploading EHR in text files 27

4.5.3 Viewing EHR data of different patients 27

4.5.4 Digital signature verification 28

4.5.5 Final output after verification 28

5. Experimental Results 29

5.1. Experiment setup 29

5.1.1. Setup Jupyter Notebook 29

5.1.2 Setting Up Visual Studio Code 30

5.2. Libraries used 32

5.2.1 Hash Library 32

5.2.2 Euclidean Algorithm 32

5.2.3 Generating Random Prime 32

5.2.4 Rabin-Miller Test 32

5.2.5 Multiplicative Inverse 33

5.2.6 Hex Digest 33

5.2.7 SHA-256 33

5.3. Parameters

5.3.1 Encryption Strength 33

5.3.2 Decentralization 34

5.4. Parameters and Formulae 34

6. Discussion of Results 34

6.1. Speech/text to Sign Language 35

7. Summary, Conclusion and Recommendation 36

8. Future Enhancements 37

9. References 38

List of Figures

Figure No.	Figure Name	Page No.
1.1	Data Breaches	2
1.2	Basic encryption and decryption of data	4
1.3	Key features of Block chain	5
1.4	Working of Public and Private keys	6
1.5	Illustration of RSA Algorithm	7
3.3.6.1	Blockchain Overview	16
4.2.4.1	Overview of transition of EHR	19
4.3.1	RSA key generation	20
4.5.1	login page for patient	26
4.5.2	Uploading EHR in text files	27
4.5.3	Viewing EHR data/Appointments of different patients	27
4.5.4	Digital signature verification	28
4.5.5	Final output after verification	28
5.2	Visual Studio Code	29
5.3.2.1	Consensus Algorithm	33
6.1	Increase in security level (using RSA)	34
6.2	Number of insertions in a block per second	36

List of Tables

Table No.	Table Name	Page No.
2.1	Comparison of Existing Method from selected Strategies	11
5.3	Parameter comparison table between existing methods and proposed methods	26
6.1	Increase in the security level when using RSA and other methodologies	35
6.1	comparison of security between existing methods and proposed methods for transmission of information	35

List of Abbreviations

Abbreviations	Full Form
DHL	Digital Health Ledger
RSA	Rivest Shamir Adleman
SHA	Secure Hash Algorithm
C(text)	Cipher text
M(text)	Plain text
E	Encrypted
D	Decrypted
Hash lib	Hash function library
B(representation)	Block representation in Block chain
UI	User interface
PY	Python
API	Application Program Interface

1. Introduction

In an era defined by digital transformation, healthcare organizations are increasingly turning to technology to enhance the management and security of patient health records. The need for efficient, secure, and scalable solutions has never been greater, and this imperative led to the development of our project – "Dynamic Health ledger." In this project, we have leveraged RSA key exchange and blockchain technology to create a robust system that addresses these critical requirements.

The secure and efficient exchange of health records across a distributed network of healthcare providers and institutions is a formidable challenge. Ensuring patient data privacy and integrity is paramount. The process we are presenting in this review aims to offer a high-performance solution to these issues by combining two powerful technologies: RSA encryption for secure key exchange and blockchain for data immutability and traceability.

Our project harnesses the power of RSA cryptography, one of the most widely used asymmetric encryption algorithms in the world. RSA is known for its robust security, and it plays a pivotal role in securing the exchange of cryptographic keys within our system. In addition to RSA encryption, we integrate blockchain technology into our distributed health records database protocol. Blockchain, the underlying technology of cryptocurrencies like Bitcoin, provides a decentralized, tamper-resistant ledger for recording and storing healthcare transactions

Throughout this project, we have delved into the technical details of our protocol, showcasing how RSA key exchange and blockchain technology work together seamlessly to create a performant, secure, and efficient system for managing distributed health records. Our aim is to provide healthcare institutions with a blueprint for implementing a secure and scalable solution that can significantly improve patient data management and ultimately enhance the quality of healthcare delivery

1.1. Motivation

Healthcare data breaches are on the rise, posing significant threats to individuals and the healthcare industry. These breaches can result in misdiagnosis, treatment delays, and eroded patient trust. According to the International Data Corporation (IDC), they also come at a staggering financial cost, with billions of dollars lost annually.

Our project, Dynamic Health Ledger (DHL), addresses these critical concerns using RSA encryption and blockchain. Much like deep learning prevents industrial accidents, our approach aims to revolutionize healthcare data management and protect patient records.

We use RSA for secure key exchange, bolstering health record protection, ensuring only authorized users gain access. Blockchain technology further secures data by creating an unchangeable record of transactions and access, rendering tampering practically impossible.

Our approach is transformative, shifting from reacting to data breaches to preventing unauthorized access. By harnessing RSA and blockchain, we not only safeguard patient data but also rebuild trust in healthcare systems, potentially saving organizations and the industry from the financial and reputational fallout of data breaches.

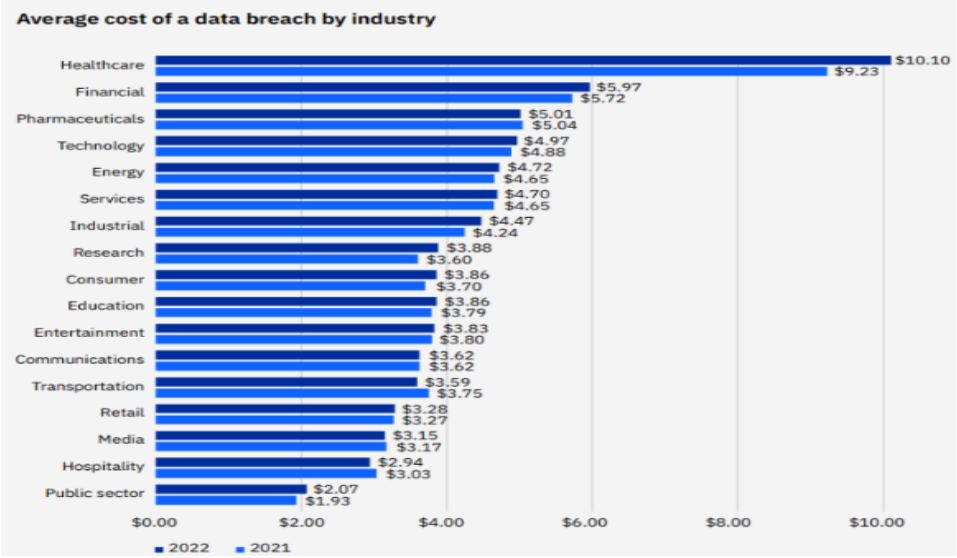

Figure 1.1: Data Breaches

1.2. Problem definition

In healthcare, ensuring the secure access and efficient management of patient data poses a significant challenge. Traditional methods often struggle to balance the need for robust security measures with the imperative of maintaining accessibility and cost-effectiveness. To tackle these obstacles, our project proposes a novel solution harnessing the power of RSA encryption for secure key exchange and blockchain technology for streamlined data management within distributed health records. RSA encryption forms the bedrock of our security strategy, providing a robust mechanism for safeguarding patient data. By utilizing RSA for secure key exchange, we establish a secure framework that enables encryption of health records, ensuring that only authorized personnel can access sensitive information.

1.3. Problem Illustration

The Dynamic Health Ledger, employing RSA for secure key exchange and blockchain technology, holds significant promise in revolutionizing healthcare data management. This innovative approach ensures the utmost security and data integrity within the health sector, safeguarding sensitive patient information. By leveraging RSA encryption, it establishes a robust foundation for secure communication and access control, while the integration of blockchain provides an immutable ledger for transparent and auditable health record transactions. The potential for enhanced interoperability, trust, and privacy protection makes this project a valuable contribution to the future of healthcare information systems.

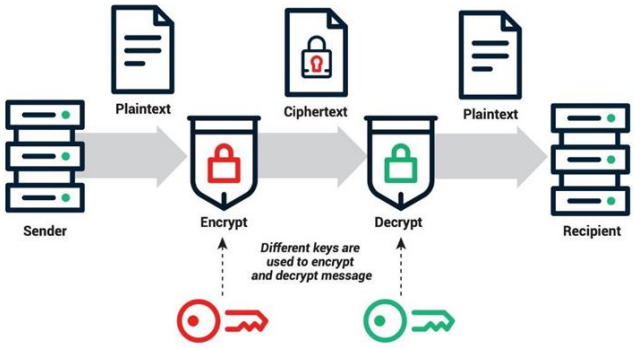

fig1.2: Basic encryption and decryption of data

Block Chain Technology

Blockchain's decentralized architecture offers resilience against single points of failure, enhancing system reliability and reducing vulnerabilities to cyberattacks. By distributing data across a network of nodes and employing consensus mechanisms to validate transactions, blockchain mitigates the risk of data manipulation or censorship. This inherent resilience makes blockchain particularly appealing for applications requiring high levels of security and trust, such as critical infrastructure, identity management, and digital asset custody. As organizations increasingly recognize the value of decentralization in safeguarding sensitive information and ensuring continuous operation, blockchain technology continues to evolve as a cornerstone of secure, trustworthy digital ecosystems.

It is known for the decentralized ledger system, records transactions securely and transparently across a network of nodes. Utilizing cryptographic techniques, it ensures data integrity and resists tampering, fostering trust among participants without the need for intermediaries. Smart contracts automate agreements, reducing costs and dispute risks. Beyond finance, blockchain finds applications in supply chains, healthcare, and voting systems, promising efficiency and security enhancements across industries.

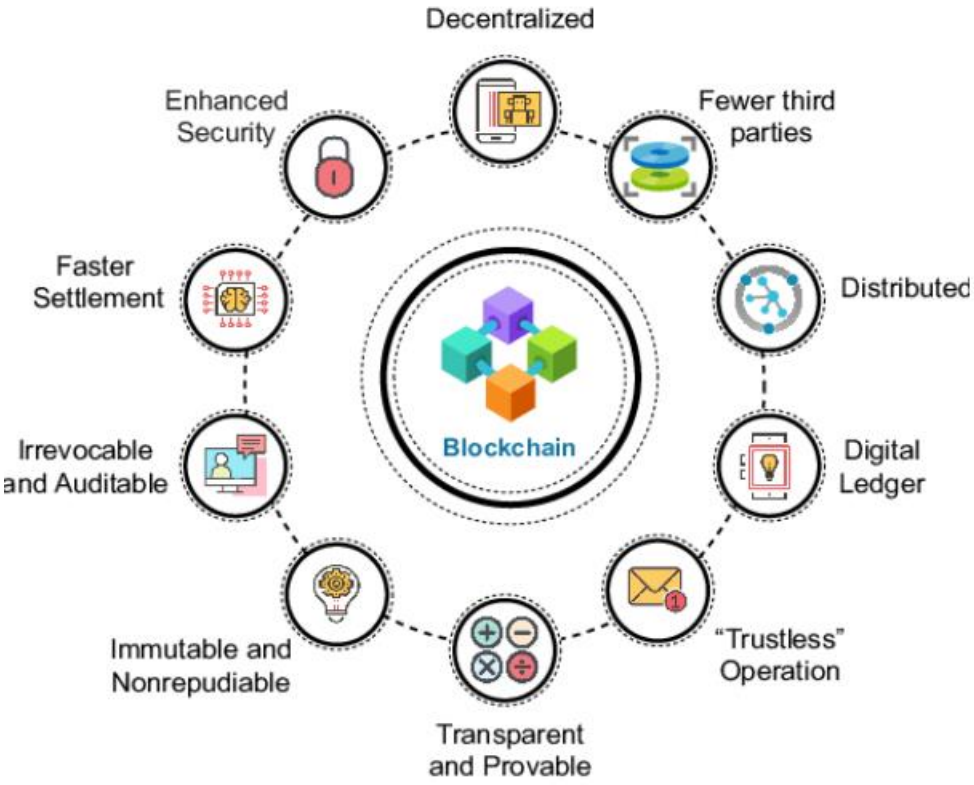

fig1.3: Key features of Block chain

Public and Private Keys:

In cryptography, public and private keys form the cornerstone of secure communication and data protection. Each key plays a distinct role in the encryption and decryption processes, ensuring confidentiality and authenticity in digital transactions. Public keys are openly shared and used for encryption, allowing anyone to send encrypted messages to the holder of the corresponding private key. This asymmetric encryption scheme enables secure communication over insecure channels, as only the recipient possessing the private key can decrypt and access the original message. The mathematical relationship between public and private keys, typically established through algorithms like RSA (Rivest-Shamir-Adleman), guarantees the confidentiality of sensitive information while facilitating seamless data exchange in digital environments.

Conversely, private keys are closely guarded secrets known only to their respective owners, providing authentication and non-repudiation in cryptographic operations. Used for decrypting messages encrypted with the corresponding public keys, private keys ensure that only authorized individuals or entities can access and decipher sensitive data. The security of private keys is paramount, as their compromise could lead to unauthorized access, data breaches, and identity theft. Cryptographic protocols and secure key management practices are employed to safeguard private keys against theft, tampering, and exploitation, reinforcing the foundation of trust and security in digital communications and transactions.

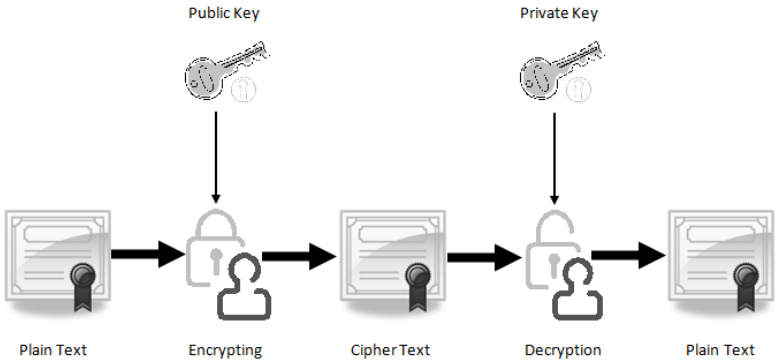

fig1.4: Working of Public and Private keys

RSA Algorithm

The RSA algorithm, named after its inventors Rivest, Shamir, and Adleman, is a cornerstone of modern cryptography renowned for its strength and versatility. Operating on the principles of asymmetric encryption, RSA utilizes a pair of keys: a public key for encryption and a private key for decryption. The security of RSA rests on the computational complexity of factoring large prime numbers, a task believed to be infeasible with current computing capabilities. This asymmetry enables secure communication channels, where sensitive information encrypted with the public key can only be decrypted by the corresponding private key, ensuring confidentiality and privacy in digital transactions. Moreover, RSA serves as the foundation for secure key exchange protocols, facilitating the establishment of secure communication channels over untrusted networks, thereby safeguarding data integrity and preventing unauthorized access to sensitive information. Its widespread adoption across various domains, including e-commerce, financial transactions, and communication networks, underscores its significance in modern cryptographic protocols as a robust and reliable encryption mechanism.

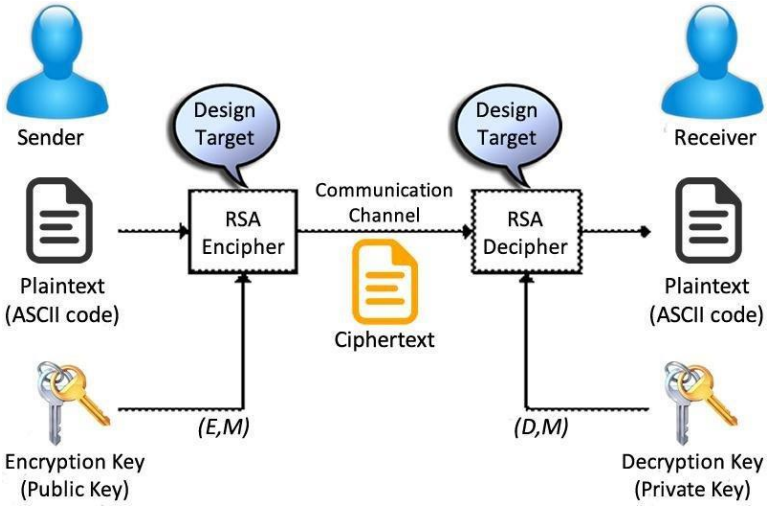

fig1.5: Illustration of RSA Algorithm

1.4. Objective

Imagine The Dynamic Health Ledger seeks to enhance the performance and efficiency of managing Electronic Health Records (EHRs) across diverse organizations. It prioritizes improving the speed and accuracy of EHR insertion, especially in scenarios involving sizable medical imaging data. The protocol aims to optimize data retrieval, minimize contention, and identify unauthorized access, all while ensuring compliance

2. Literature Survey

The existing system for healthcare records management varies widely depending on the healthcare provider, region, and technological infrastructure. However, some common characteristics and challenges in existing healthcare records management systems include:

- **Paper-Based Records:** Many healthcare providers still rely on paper-based records, which are cumbersome to manage, prone to physical damage, and can be challenging to access and share securely.
- **Electronic Health Records (EHRs):** In more technologically advanced settings, Electronic Health Records (EHRs) have been adopted. EHRs are digital versions of patient medical records, but interoperability issues often limit their effectiveness. Different systems may use incompatible standards and formats, making it difficult to share information across different healthcare institutions.
- **Centralized Databases:** Some healthcare providers maintain centralized databases for storing patient records. These databases are vulnerable to single points of failure, data breaches, and can be challenging to scale to accommodate a growing volume of data.
- **Limited Patient Control:** Patients often have limited control over their own health records and may face challenges accessing and sharing their information with other healthcare providers when needed.
- **Data Redundancy:** Inefficiencies can arise from redundant data entry and maintenance, which can lead to errors and duplication of efforts.
- **Limited Data Sharing:** Sharing healthcare records with other authorized healthcare providers, especially in emergency situations, can be slow and inefficient, impacting patient

3. Proposed Method

The proposed system, “Dynamic Health Ledger” harnesses the power of RSA cryptography, one of the most widely used asymmetric encryption algorithms in the world. RSA is known for its robust security, and it plays a pivotal role in securing the exchange of cryptographic keys within our system. In addition to RSA encryption, we integrate blockchain technology into our distributed health records database protocol. Blockchain, the underlying technology of cryptocurrencies like Bitcoin, provides a decentralized, tamper-resistant ledger for recording and storing healthcare transactions. Certainly, here are some potential applications and use cases related to the proposed system in the which aims to detect various safety-related factors in real time:

- **Data Security:** RSA encryption provides a robust and secure method for exchanging encryption keys, ensuring that sensitive health records remain confidential. Utilizing blockchain technology enhances data security by creating an immutable ledger where once data is recorded, it cannot be altered. This ensures data integrity and trust
- **Data Integrity:** Blockchain's design ensures that once data is recorded, it cannot be tampered with or altered. This is crucial for maintaining the integrity of health records, preventing unauthorized changes.
- **Decentralization:** Blockchain is inherently decentralized, meaning that there's no single point of failure. Health records are stored across multiple nodes, reducing the risk of data loss or unauthorized access
- **Data Ownership and Consent:** Blockchain can enable patients to have greater control over their own health data. They can grant or revoke access to their records through smart contracts, ensuring their consent is always obtained
- **Redundancy and Disaster Recovery:** With distributed data across multiple nodes, your system is more robust against data loss due to hardware failures or natural disasters.

3.1 ANALYSIS

Analyzing the Performant Protocol for Distributed Health Records Databases, which leverages RSA for key exchange and blockchain technology, for your final project review involves assessing the protocol's design, security, and performance aspects. Here's an analysis of these key elements:

3.1.1 RSA Key Exchange: RSA is a widely-adopted asymmetric encryption algorithm, known for its security and mathematical robustness. Using RSA for key exchange ensures that sensitive health records' encryption keys are securely shared between parties. It provides strong security against eavesdropping and man-in-the-middle attacks. RSA key exchange can be computationally intensive, especially for large databases. The performance may vary based on the system's processing power and the key size used.

3.1.2 Blockchain Integration: Blockchain technology offers a tamper-proof, decentralized ledger for storing health records. This ensures data integrity and transparency. Immutability in a blockchain enhances the security and trustworthiness of the records. The decentralized nature of the blockchain can reduce the risk of a single point of failure and enhance fault tolerance.

3.1.3 Scalability and User Experience: The ability to scale the system to accommodate a growing number of health records and users is crucial. The blockchain's scalability plays a pivotal role in this aspect. A user-friendly interface and efficient access to health records are vital for healthcare providers.

The Performant Protocol for Distributed Health Records Databases, employing RSA for secure key exchange and blockchain technology, holds significant promise in revolutionizing healthcare data management. This innovative approach ensures the utmost security and data integrity within the health sector, safeguarding sensitive patient information. By leveraging RSA encryption, it establishes a robust foundation for secure communication and access control, while the integration of blockchain provides an immutable ledger for transparent and auditable health record transactions. The potential for enhanced interoperability, trust, and privacy protection makes this project a valuable contribution to the future of healthcare information systems.

This project is focused on the development and implementation of an innovative Health care records storage system that leverages efficient and immutable transactions of the records.

The scope of the project includes:

- **Technology Implementation:** The project will involve the practical application of Cryptographic key exchange techniques, specifically RSA, and also Blockchain to enhance Healthcare records safety.
- **Safety Enhancement:** The primary aim of the project is to address the pressing issue of sharing and storing Healthcare records. The focus is on preventing tampering and loosing of important data.
- **Access Control Mechanisms:** The system defines and enforces access control mechanisms to ensure that only authorized individuals and entities can access and modify health records. This includes defining roles, permissions, and authentication protocols.
- **Data Encryption:** This project includes data encryption mechanisms to protect the confidentiality of health records during storage and transmission**.**
- **Key Exchange using RSA**: The project scope includes the implementation and integration of RSA (Rivest–Shamir–Adleman) encryption for secure key exchange. RSA is used to establish secure communication channels between different nodes and participants in the distributed health records system.

- **Blockchain Integration:** The project incorporates blockchain technology for secure, transparent, and immutable record-keeping. The scope covers the design and implementation of the blockchain architecture, including smart contracts for access control and audit trail management.
- **Feasibility Demonstration:** The project's results will serve to demonstrate the practical feasibility of implementing RSA key exchange method and Blockchain technology for healthcare records safety enhancement.

3.2 RSA Implementation

3.2.1: Key Generation:

Prime Number Generation: RSA relies on the difficulty of factoring large composite numbers into their prime factors.

Prime numbers p and q are typically chosen to be large (several hundred digits long) and are generated randomly.

3.2.1.1 Modulus and Euler's Totient Function: The modulus n is calculated as the product of p and q. Euler's totient function φ(n) is used to determine the number of positive integers less than n that are coprime with n.

3.2.1.2 Public and Private Exponents: The public exponent e is typically chosen to be a small prime, often 65537 (2^16 + 1), due to its efficiency in encryption operations. The private exponent d is calculated as the modular multiplicative inverse of e modulo φ(n), ensuring that $e*d \equiv 1 \pmod{\varphi(n)}$.

3.2.2: Encryption:

3.2.2.1 Plaintext Conversion: Before encryption, the plaintext message is typically converted into a numerical format suitable for RSA encryption. This conversion process may involve techniques such as padding to ensure security and uniform message lengths.

3.2.2.2 Exponential Encryption: The encryption process involves raising the plaintext message m to the power of the public exponent e modulo the modulus n. This operation ensures that the ciphertext c is a residue class modulo n, preserving the mathematical properties necessary for decryption.

3.2.2.3 Security Considerations: RSA encryption is computationally intensive, especially for large plaintext messages. Therefore, hybrid encryption schemes are often employed, where RSA is used to encrypt a symmetric encryption key, and symmetric encryption algorithms like AES (Advanced Encryption Standard) are used to encrypt the actual message.

3.2.3 Decryption:

3.2.3.1 Exponential Decryption: Decryption involves raising the ciphertext c to the power of the private exponent d modulo the modulus n. This operation effectively reverses the encryption process, yielding the original plaintext message m.

3.2.3.2 Plaintext Reconstruction: Once the numerical representation of the plaintext message is obtained, any padding applied during encryption is typically removed to retrieve the original plaintext message.

3.2.4 Security Considerations:

Key Length: The security of RSA depends heavily on the length of the modulus n, with longer key lengths providing greater resistance against attacks. As computational power increases over time, it is essential to periodically reassess key length recommendations to maintain adequate security.

Random Number Generation: Secure random number generation is crucial for generating prime numbers and selecting cryptographic keys. Weaknesses in random number generation can lead to vulnerabilities in RSA implementations.

Padding Schemes: Proper padding schemes, such as PKCS#1 v1.5 or OAEP, are essential for RSA security. Padding mitigates attacks like chosen ciphertext attacks and ensures semantic security.

Side-Channel Attacks: RSA implementations are susceptible to side-channel attacks, where an attacker exploits information leaked through physical channels such as timing, power consumption, or electromagnetic radiation. Countermeasures like constant-time algorithms and secure hardware implementations are necessary to mitigate these attacks.

By understanding these key aspects of RSA encryption and decryption, developers can implement robust and secure RSA-based cryptographic systems for various applications, including secure communication, digital signatures, and data protection.

3.3 BLOCK CHAIN TECHNOLOGY:

Blockchain is a decentralized, distributed ledger technology that enables the secure recording and storage of transactions across a network of computers. It is most commonly known as the underlying technology behind cryptocurrencies like Bitcoin, but its applications extend far beyond digital currencies to various industries, including finance, supply chain management, healthcare, and more.

The implementation of blockchain involves several key components and processes:

3.3.1 Decentralized Network: A blockchain network consists of multiple nodes (computers) that participate in the validation and verification of transactions. Unlike traditional centralized systems where a single entity controls the database, blockchain operates on a decentralized peer-to-peer network, ensuring transparency, resilience, and censorship resistance.

3.3.2 Blocks: Transactions are grouped into blocks, which are then added to the blockchain in a chronological order. Each block contains a set of transactions, a timestamp, and a reference to the previous block, forming a continuous chain of blocks.

3.3.3 Consensus Mechanisms: To maintain the integrity of the blockchain, consensus mechanisms are employed to validate and agree on the validity of transactions before they are added to the ledger. Popular consensus algorithms include Proof of Work (PoW), Proof of Stake (PoS), and Practical Byzantine Fault Tolerance (PBFT), each with its own advantages and trade-offs.

3.3.4 Cryptography: Cryptography plays a crucial role in ensuring the security and immutability of blockchain data. Each block is cryptographically linked to the previous one, creating a tamper-evident chain where any attempt to alter past transactions would be computationally infeasible and easily detectable.

Additionally, cryptographic techniques such as digital signatures are used to authenticate transactions and verify the identity of participants.

3.3.5 Smart Contracts (optional): Some blockchain platforms, like Ethereum, support the execution of smart contracts - self-executing contracts with predefined rules and logic. Smart contracts enable the automation of complex transactions and the creation of decentralized applications (DApps) that operate autonomously without the need for intermediaries.

3.3.6 Permissioned vs. Permissionless Blockchains: Blockchain networks can be categorized as permissioned or permissionless, depending on who can participate in the network and perform certain actions. Permissionless blockchains, like Bitcoin and Ethereum, are open to anyone, allowing anyone to participate in the network, validate transactions, and add blocks to the chain. In contrast, permissioned blockchains restrict access to authorized participants, making them suitable for enterprise applications where privacy, scalability, and regulatory compliance are important consideration.

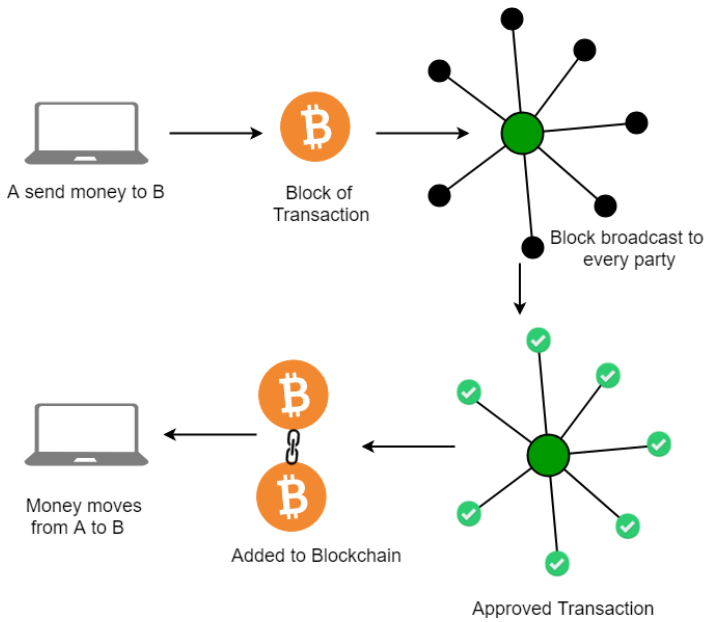

Figure: 3.3.6.1 Blockchain Overview

4. Implementation

4.1 Functionality:

4.1.1 **Patient-Controlled Access:** Through the use of RSA encryption and blockchain technology, patients can securely control access to their health records. Patients are provided with private keys generated through RSA encryption, which they can use to grant and revoke access to their medical data stored on the blockchain. This functionality ensures that patients have ultimate authority over who can view their health information, enhancing privacy protection and empowering individuals to manage their own healthcare data. Additionally, all access transactions are recorded on the immutable blockchain, providing transparency and accountability in the access management process.

4.1.2 **Interoperable Health Data Exchange:** The platform facilitates seamless and secure exchange of health data between different healthcare providers, systems, and stakeholders. Using RSA encryption for secure key exchange, authorized parties can securely share and access patient health records stored on the blockchain. This interoperability enhances collaboration among healthcare entities, allowing for comprehensive patient care and streamlined processes. Additionally, the use of blockchain technology ensures data integrity and traceability throughout the exchange process, mitigating the risk of data tampering or unauthorized access. As a result, healthcare providers can access relevant patient information efficiently and accurately, leading to improved clinical decision-making and patient outcomes.

4.1.3 **Immutable Audit Trail:** The platform maintains an immutable audit trail of all health record transactions, including access requests, modifications, and updates.

Through the use of blockchain technology, every interaction with the patient's health data is recorded in a secure and tamper-proof manner.

This audit trail provides a comprehensive history of who accessed the data, when it was accessed, and any changes made to the records. Healthcare providers, patients, and regulatory authorities can leverage this feature to ensure compliance with privacy regulations, track data usage.

And investigate any discrepancies or unauthorized access attempts. By maintaining a transparent and immutable record of data transactions, the platform enhances trust and accountability in the management of healthcare information.

The implementation of the project has been carried out in a step-by-step manner. A detailed description of each module is given below and it is followed by an introduction to the technologies used in implementing the project.

4.2 SYSTEM ARCHITECTURE AND METHODOLOGY

The system's architecture should be robust and secure to ensure the integrity and confidentiality of health records. It should include several key components and layers:

4.2.1 User Interface Layer: This layer is the user-facing part of the system and includes web or mobile applications for patients, healthcare providers, and administrators. Users can authenticate themselves using their credentials or biometrics. This layer interacts with the Application Layer.

4.2.2 Application Layer: This layer contains the core business logic of the system, handling user requests and orchestrating various functionalities.

Key components in this layer include**:**

1.Authentication and Authorization: Verify user identities and manage access control.

2. Health Record Management: Create and retrieve health records.

3.Blockchain Integration: Add health records to the blockchain and verify data integrity.

4.RSA Key Management: Generate and manage RSA keys for encryption and decryption.

4.2.3 Blockchain Layer: This layer is responsible for secure and immutable storage of health records. It leverages blockchain technology for decentralized and tamper-proof record keeping. Health records are hashed and added to the blockchain, ensuring data integrity and auditability. Access to this layer is controlled by smart contracts, which enforce rules and permission.

4.2.4 Security Layer: This layer is dedicated to ensuring the security of the system, focusing on data encryption and secure key exchange.

1.RSA Key Exchange: Secure communication channels between users are established using RSA key exchange protocols.

2.Encryption: Health records are encrypted before transmission and storage, and only authorized users possess the keys for decryption.

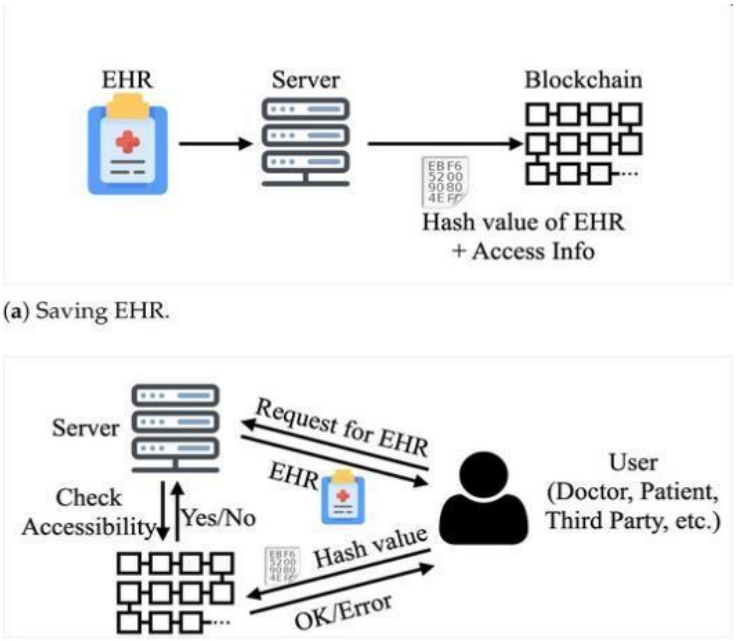

Figure: 4.2.4.1 Overview of transition of EHR

4.3 RSA KEY GENERATION:

In the "Performant Protocol for Distributed Health Records Databases," the RSA key exchange algorithm is employed to establish secure communication channels between users, ensuring the confidentiality and integrity of health records. When two parties, such as a healthcare provider and a patient, need to communicate securely, the system generates a pair of RSA keys for each party: a public key and a private key. The public keys are exchanged, while the private keys are kept securely by the respective parties. Messages sent by one party are encrypted with the recipient's public key, ensuring that only the recipient, holding the corresponding private key, can decrypt and access the information. This asymmetric encryption scheme provides a strong level of security, allowing for confidential and authenticated communication while mitigating the risk of eavesdropping and data tampering. The RSA key exchange algorithm is a fundamental component of the system's security infrastructure, safeguarding sensitive health data during transmission.

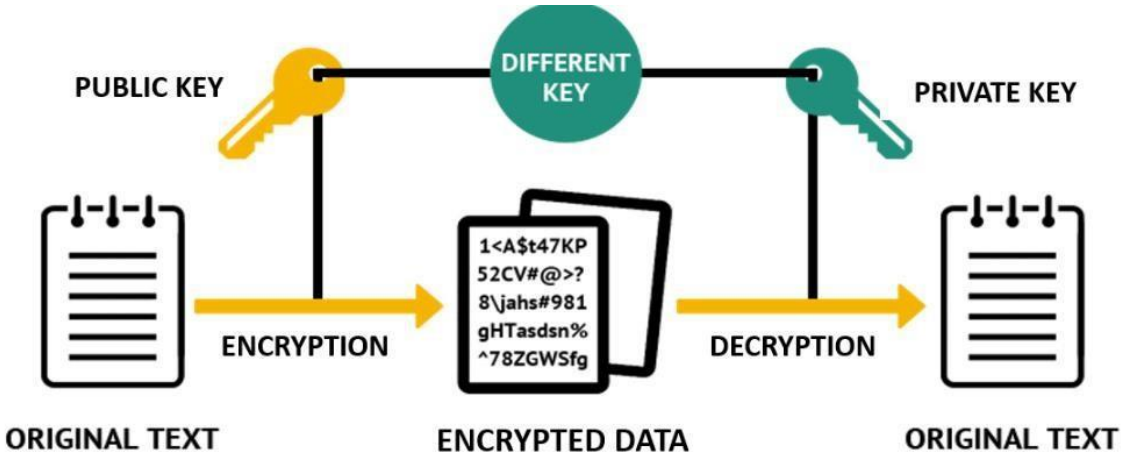

Figure: 4.3.1 RSA KEY GENERATION

4.3.1 Encryption with Public Key: The sender encrypts the session key using the recipient's public key. This encrypted session key is then transmitted securely to the recipient.

4.3.2 Decryption with Private Key: The recipient uses their private key to decrypt the session key sent by the sender. This action ensures that only the recipient can access the session key.

4.3.3 Secure Communication: With the session key established, both parties can securely encrypt and decrypt messages using symmetric encryption algorithms, such as AES. This guarantees the confidentiality and integrity of their communication.

4.4 BLOCKCHAIN IMPLEMENTATION:

In the "Performant Protocol for Distributed Health Records Databases," the blockchain is implemented as a foundational technology to enhance the security and integrity of health record data. A private or consortium blockchain network is employed to ensure privacy and control over data access. Each health record update is hashed and added as a transaction to the blockchain. Smart contracts govern access control, allowing only authorized entities such as healthcare providers and administrators to interact with the blockchain layer. This ensures that only approved and authenticated users can append records, enhancing data integrity and security. The blockchain's decentralized, tamper-proof ledger provides a transparent and immutable record of all health record changes, enabling reliable audit trails and compliance with healthcare data regulations. Overall, the blockchain implementation in

this project significantly enhances the trustworthiness of health records and ensures the confidentiality and reliability of sensitive patient information.

4.4.1 Smart Contract Development: Create smart contracts that define the rules and permissions for interacting with the blockchain. Implement functions for adding health records, verifying data integrity, and managing access control.

4.4.2 Data Verification: Users, including patients and healthcare providers, can verify the integrity of health records by cross-referencing the blockchain's stored hashes with the locally stored data.

4.4.3 Access Control: Implement an access control mechanism using the smart contracts to ensure that only authorized users, such as healthcare providers and patients, can interact with the blockchain. Define roles and permissions within the smart contracts.

4.4.4 SAMPLE CODE

4.4.4.1 main.py

```
import hashlib class

GeekCoinBlock:

    def __init__(self, previous_block_hash, transaction_list):

        self.previous_block_hash = previous_block_hash

    self.transaction_list = transaction_list

self.block_data = f"{transaction_list} - {previous_block_hash}"

self.block_hash = hashlib.sha256(self.block_data.encode()).hexdigest()

def generateblockchain(blockc,f1):

 block1 = GeekCoinBlock(blockc,f1)

  datas=block1.block_data

   haskey=block1.block_hash

print(f"Block 1 data: {block1.block_data}")

   print(f"Block 1 hash: {block1.block_hash}")

 return datas,haskey
```

4.4.4.2 RSA.py

```
 import random from fractions import
gcd

def extendedEuclidean(num1, num2):

   if num2 == 0:

       return (num1, 1, 0)
d, temp_x, temp_y = extendedEuclidean(num2, num1 % num2)
 x, y = temp_y, temp_x - int(num1 / num2) * temp_y
    return (d, x, y) def
rabinMillerTest(p, iteration):
if p < 2:
       return False
if p != 2 and p % 2 == 0:
       return False
 s = p-1
while s % 2 == 0:
       s //= 2
 for i in range(iteration):
 a = random.randint(1, p-1)
  temp = s
  mod = pow(a, temp, p)
# Computes (a^temp)%p
   while temp != p-1 and mod != 1 and mod != p-1:
```

```
            mod = pow(mod, mod, p)
temp *= 2
 if mod != p-1 and temp % 2 == 0:
            return False
 return True
 def multiplicativeInverse(a, b, n):
    d, x, y = extendedEuclidean(a, n)
   if b % d == 0:
 temp_x = (x * (b/d)) % n
result = []
for i in range(d):
result.append((temp_x + i*(n/d)) %
n)         return result
 return []
def generateRandomPrime(bits):
 num = random.getrandbits(bits - 1)
  if num % 2 == 0:
   num -= 1
num += (1 << (bits - 1))
  while(not rabinMillerTest(num, 40)):
        num += 2     return
num def
generate(bits=512):
p = generateRandomPrime(bits//2)
q = p
 while q == p:
```

```
    q = generateRandomPrime(bits//2)
 n = p*q
phi = (p-1) * (q-1)
e = random.randint(1, 50000)
 e = 2*e + 1
while not (gcd(phi, e) == 1):
        e = random.randint(1, 50000)
 e = 2*e + 1
 d = multiplicativeInverse(e, 1, phi)[0]
return {
        "public": (e, n),
        "private": (int(d), n)
    }
def encrypt(keys, text):
    key, n = keys
 result = [pow(ord(c), key, n) for c in text]
return result def decrypt(keys, text):
    key, n = keys
result = [chr(pow(int(c), int(key), int(n) )) for c in text]
return ''.join(result)
```

4.5 Experimental Screenshots:

4.5.1 login page for patient

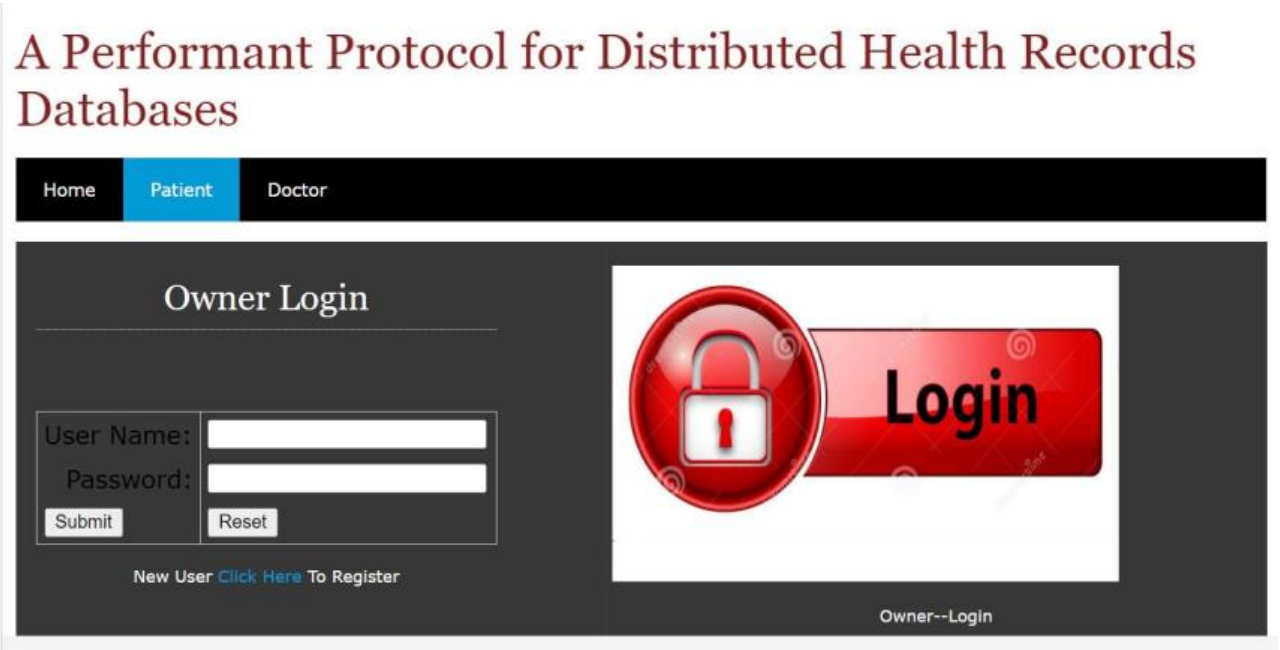

Figure: 4.5.1 login page for patient

4.5.2 Uploading EHR in text files

Figure: 4.5.2 Uploading EHR in text files

4.5.3 Viewing EHR data/Appointments of different patients

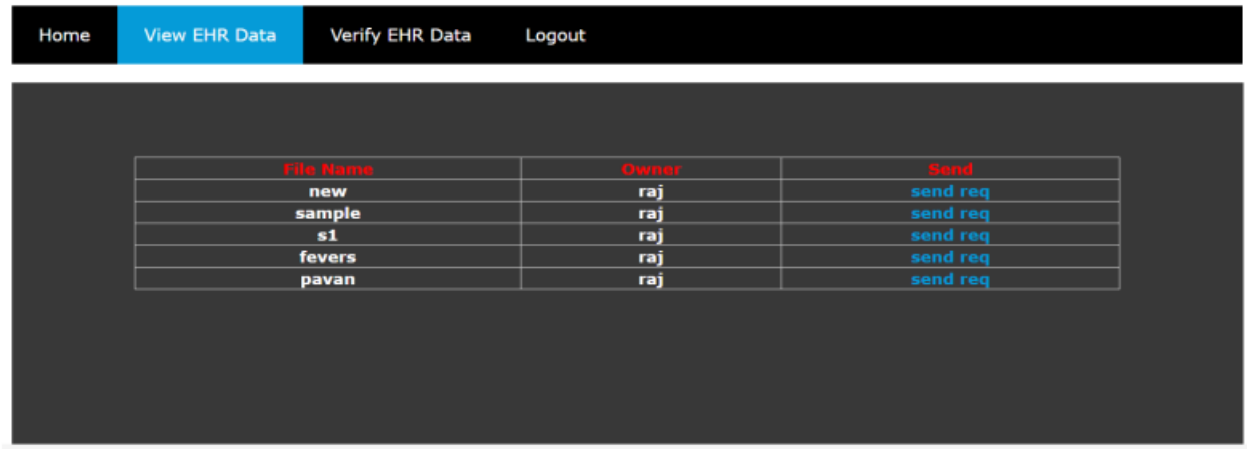

Figure: 4.5.3 Viewing EHR data/Appointments of different patients

4.5.4 Digital signature verification

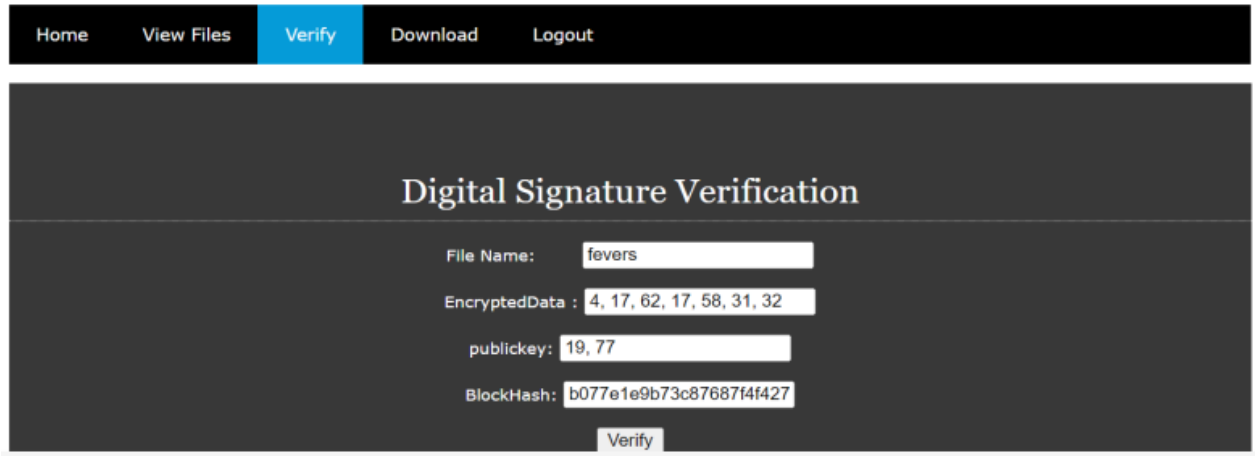

Figure: 4.5.4 Digital signature verification

4.5.5 Final output after verification

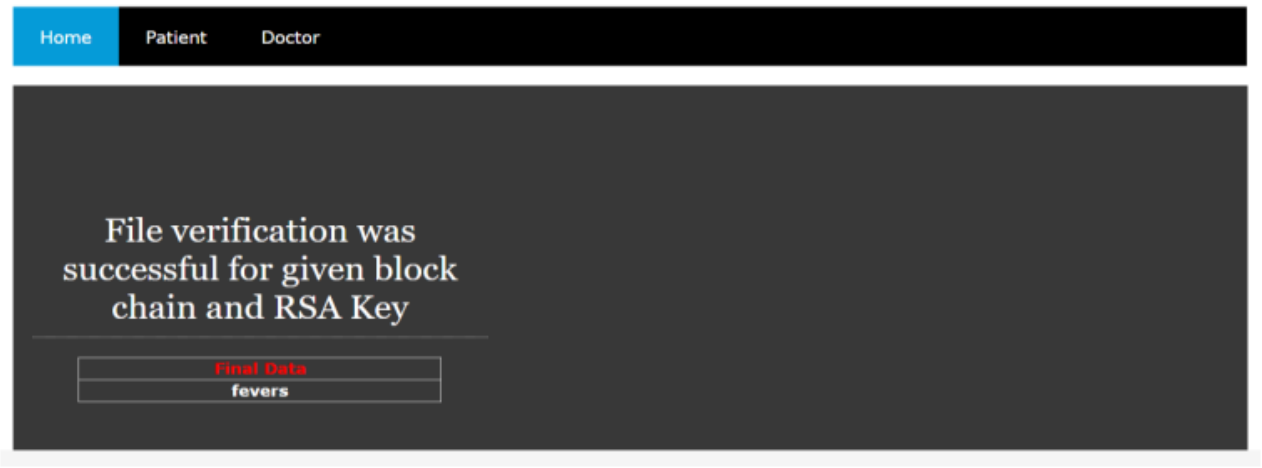

Figure: 4.5.5 Final output after verification

5. Experimental Results

5.1 Experimental setup:

5.1.1. Setup Jupyter Notebook:

To install and set up Jupyter Notebook, you can follow these steps:

1. **Install Python:** First, you need to have Python installed on your system. You can download and install Python from the official Python website: https://www.python.org/. Make sure to check the option to add Python to your system PATH during installation.

2. Install Jupyter Notebook: Once Python is installed, you can install Jupyter Notebook using pip, which is the Python package manager. Open a terminal or command prompt and run the following command: **pip install jupyter**

3. Launch Jupyter Notebook: After the installation is complete, you can launch Jupyter Notebook by running the following command in your terminal or command prompt: **jupyter notebook**

4. **Accessing Jupyter Notebook**: Once you run the command, your default web browser should open, and you'll be directed to the Jupyter Notebook dashboard. If it doesn't open automatically, you can manually open your web browser and go to http://localhost:8888/. Here, you'll see a file browser where you can navigate your filesystem and create or open Jupyter Notebook files.

5. **Creating a New Notebook**: To create a new notebook, click on the "New" button in the top right corner and select "Python 3" (or any other available kernel you want to use).

6. Using Jupyter Notebook: You can now start using Jupyter Notebook. Each notebook consists of cells where you can write and execute Python code, Markdown for documentation, and more.

7. Saving and Closing: Make sure to save your work regularly by clicking the "Save" button or using the keyboard shortcut Ctrl+S. To close Jupyter Notebook, you can simply close the browser tab or stop the Jupyter Notebook server by pressing Ctrl+C in the terminal or command prompt where it's running.

5.1.2. Setting Up Visual Studio Code

Here's a guide to installing and configuring Visual Studio Code (VS Code) for a smooth development experience:

1. Download and Install: Head to the official VS Code download page: https://code.visualstudio.com/download Choose the installer suitable for your operating system (Windows, Mac, or Linux). Run the downloaded installer and follow the on-screen instructions.

2. Open VS Code: Once installed, locate and launch VS Code from your applications list.

3. Install Extensions (Optional): VS Code is powerful with extensions. Explore the Extensions marketplace within VS Code and install language-specific extensions for syntax highlighting, code completion, and debugging support relevant to your project needs.

4. Open Your Project: Navigate to your project folder using the File Explorer within VS Code (File > Open Folder).

You're now ready to start coding! Use the built-in features like syntax highlighting, code completion, and debugging to write and test your code efficiently.

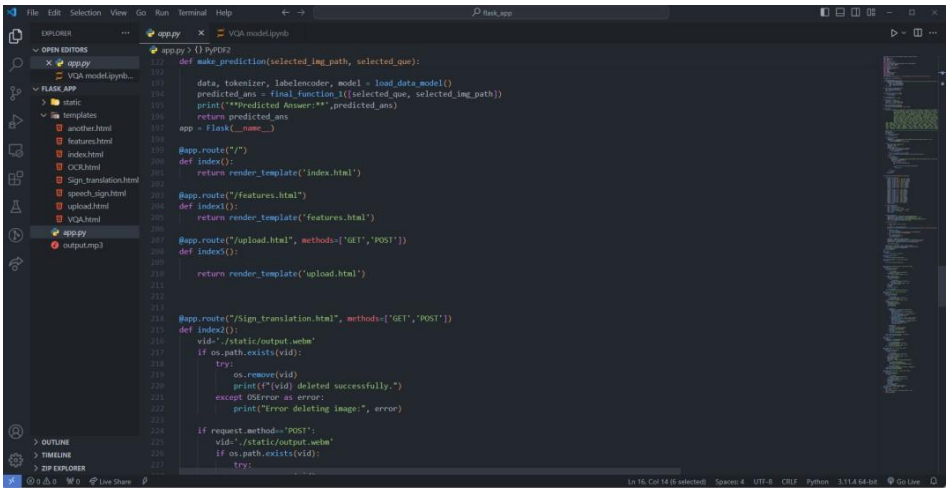

Fig 5.2: Visual Studio Code

5. Install Extensions (Optional): VS Code is powerful with extensions. Explore the Extensions marketplace within VS Code and install language-specific extensions for syntax highlighting, code completion, and debugging support relevant to your project needs.

6. Open Your Project: Navigate to your project folder using the File Explorer within VS Code (File > Open Folder).

You're now ready to start coding! Use the built-in features like syntax highlighting, code completion, and debugging to write and test your code efficiently.

5.2. Libraries Used:

5.2.1 Hash Library:

A hash function is a mathematical algorithm that converts an input (or 'message') into a fixed-size string of bytes. Hash functions are commonly used in cryptography and data integrity applications. Libraries such as hashlib in Python provide implementations of various hash functions like SHA-256 and MD5. These libraries allow users to compute the hash value of data and verify its integrity.

5.2.2 Euclidean Algorithm:

The Euclidean algorithm is used to find the greatest common divisor (GCD) of two integers. It works by iteratively applying the principle that the GCD of two numbers is the same as the GCD of the smaller number and the remainder of the division of the larger number by the smaller number. This process continues until the remainder is 0. The last non-zero remainder is the GCD of the original two numbers.

5.2.3 Generating Random Prime:

Generating a random prime number involves selecting a random integer and then testing if it is prime. One common approach is to use probabilistic primality tests like the Rabin-Miller test to quickly determine if a number is likely prime. If the number passes the test, it is considered a prime candidate. If not, another random number is selected and tested until a prime is found.

5.2.4 Rabin-Miller Test:

The Rabin-Miller primality test is a probabilistic algorithm used to determine if a given number is likely prime. It works by repeatedly choosing random bases and checking if the number passes certain conditions based on modular arithmetic. If the number passes a sufficient number of iterations with different bases, it is considered likely prime with a high degree of confidence.

5.2.5 Multiplicative Inverse:

In modular arithmetic, the multiplicative inverse of an integer 'a' modulo 'm' is another integer 'b' such that

$(a \times b) \bmod m = 1$. It is denoted as a to the power of -1. Finding the multiplicative inverse is essential in RSA cryptography for generating the private key from the public key components.

5.2.6 Hex Digest:

A hex digest is a hexadecimal representation of the output of a hash function. Hash functions typically produce binary output, but hex digests are easier to read and work with in many contexts. For example, the SHA-256 hash function produces a 256-bit binary output, which can be represented as a 64-character hexadecimal string.

5.2.7 SHA-256:

SHA-256 (Secure Hash Algorithm 256-bit) is a widely used cryptographic hash function that produces a 256-bit (32-byte) hash value. It takes an input message of any length and produces a fixed-size output hash. SHA-256 is commonly used in various cryptographic applications such as digital signatures, message authentication codes, and blockchain technology for its strong collision resistance properties.

for foundational tasks, consider Scikit-image for more advanced image processing applications.

5.3. Parameters:

5.3.1 **Encryption Strength:** Evaluate the strength of the encryption algorithm (e.g., RSA) used to encrypt the electronic healthcare records (EHR) and digital signatures.

Data Integrity: Assess the mechanisms in place to ensure the integrity of the data stored blockchain and during transmission.

Access Control: Examine the effectiveness of access control mechanisms in regulating access to patient data and preventing unauthorized access.

5.3.2 Decentralization:

Blockchain Consensus: Evaluate the decentralization achieved through the blockchain network consensus mechanism (e.g., proof of work, proof of stake).

Node Distribution: Assess the distribution of blockchain nodes across different geographical locations to ensure decentralization and fault tolerance.

Data Replication: Examine how data replication is managed across the decentralized network to ensure redundancy and availability.

User Experience: Account Creation: Evaluate the ease of account creation process for patients and doctors.

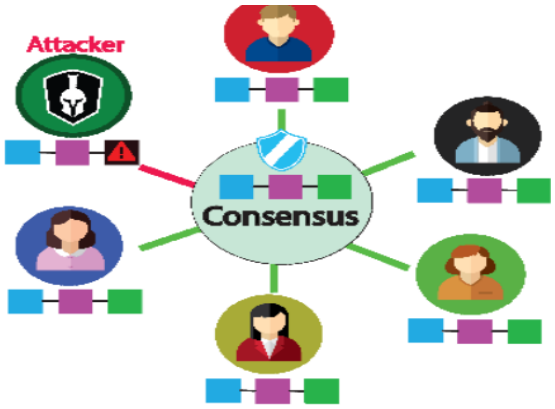

Figure 5.3.2.1 Consensus mechanism

Parameter	Previous methods	Proposed method
Email Technology	In the previous methods abstain from utilizing email technology as a means for transmitting public and private keys.	This approach integrates email technology as a means for transmitting public and private keys securely.
RSA Utilization	In the previous methods it relies on alternative security measures for ensuring data integrity and access control.	The updated version employs RSA for ensuring data integrity and access control, leveraging its cryptographic capabilities alongside alternative security measures.
Time	In the previously used method, the generation time of video is low.	The generation time of video is fast as we load the videos from the existence.
Security	In the previously used method, they provided security to some extent.	In this used method, we are providing security more than that with the help of RSA algorithm as well as blockchain technology.

Table 5.3.2.2: Parameter comparison table between existing methods and proposed method

5.4 PARAMETERS AND FORMULAE:

Mathematical Formulae:

RSA Key Generation Formulas:

1.Select Two Large Prime Numbers (p and q):

- Choose two distinct prime numbers, p and q.

2.Calculate the Modulus (n):

- Compute the modulus (n) as the product of p and q: $n = p * q$

3. Calculate Euler's Totient Function (φ(n)):

- Calculate φ(n), the totient function of n, as follows:

$$\varphi(n) = (p - 1) * (q - 1)$$

1. Select an Encryption Exponent (e):

- Choose an encryption exponent (e) such that $1 < e < \varphi(n)$ and e is relatively prime to φ(n). Common choices for e include 3 or 65537.

2. Calculate the Decryption Exponent (d):

- Calculate the modular multiplicative inverse of e modulo φ(n). This is often done using the Extended Euclidean Algorithm. The formula is:

$$d \equiv e^{-1} \pmod{\phi(n)}$$

The public key consists of (n, e), and the private key consists of (n, d).

Here are some key variables:

- n: The modulus, used in both the public and private keys.
- e: The public exponent, used for encryption.
- d: The private exponent, used for decryption.
- p and q: The prime factors of the modulus n.
- φ(n): Euler's totient function of n.

6. Discussion of Results

The results of the project showcase a robust system for securely storing and accessing electronic healthcare records (EHRs) through the innovative use of blockchain technology and cryptographic techniques. By decentralizing data storage and employing encryption, the platform ensures high levels of data security and integrity, safeguarding patient information from unauthorized access and tampering. However, while the security measures implemented add layers of protection, they also introduce complexities in the user experience. The multi-step process for patients to approve access requests and share cryptographic keys may lead to usability issues and potential delays for healthcare providers seeking access to patient data. Balancing security with usability remains a crucial consideration for the platform's effectiveness in real-world healthcare settings. Furthermore, scalability and performance are areas of concern, particularly as the user base expands. Optimization of system architecture and cryptographic algorithms is essential to mitigate potential bottlenecks and ensure efficient operation as demand grows. Overall, while the project demonstrates the feasibility of leveraging blockchain and cryptography for healthcare data management, ongoing refinement and evaluation are necessary to address usability, scalability, and performance challenges for broader adoption and impact in the healthcare industry.

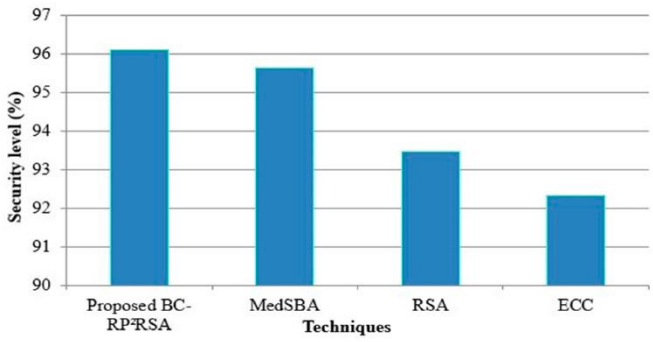

Figure 6.1. Increase in security level (using RSA)

6.1. Text/Speech to Sign:

	Existing Method	Proposed Method
Security	**80%**	**100%**

Graph:

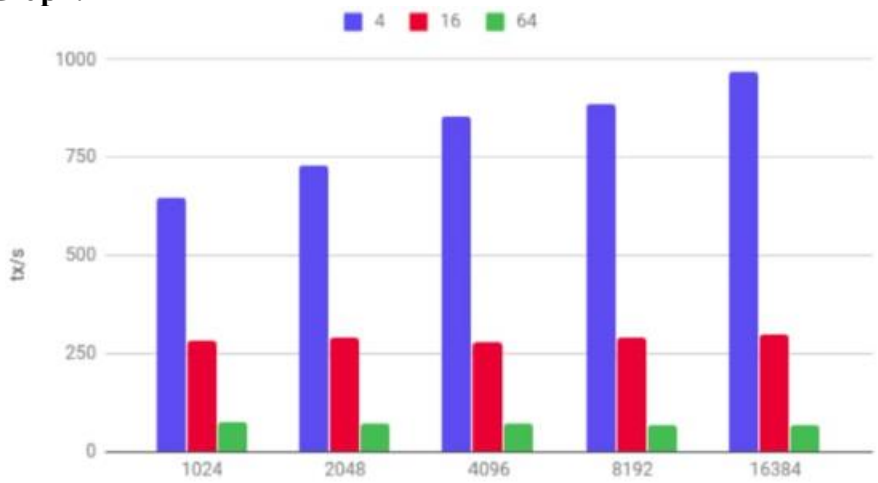

Figure 6.2. Number of insertions in a block per second

One of the greatest strengths of blockchain technology lies in its ability to rapidly add and insert values into a block within a short period of time. This efficiency is attributed to the decentralized and distributed nature of blockchain networks, which operate on consensus mechanisms that enable quick validation and verification of transactions. When a new value needs to be added to a block, it undergoes a process of validation by network participants, known as miners or validators, who compete to solve complex mathematical puzzles. Once a solution is found and verified by the majority of nodes in the network, the new block is appended to the existing blockchain in a matter of minutes. This streamlined process ensures near-instantaneous addition of values to the blockchain, facilitating real-time transactions and data updates across the network. Furthermore, the transparency and immutability of blockchain records provide assurance of data integrity and security, making it a trusted platform for a wide range of applications, including finance, supply chain management, and healthcare.

.

7. Summary, Conclusion and Recommendation

The project aims to provide a secure and decentralized platform for storing and accessing electronic healthcare records (EHRs) of patients. Utilizing blockchain technology and encryption techniques, the platform ensures data security, integrity, and accessibility for patients, doctors, and administrators. Patients can securely upload their EHRs to the blockchain, along with encrypted disease text files and digital signatures. When doctors request access to a patient's EHR, the patient approves the request and provides the necessary digital signature and public key for verification. Once verified, doctors can decrypt the disease text file, diagnose the patient, and provide appropriate medical treatment.

The project successfully demonstrates the feasibility of utilizing blockchain and encryption technologies to enhance the security and accessibility of electronic healthcare records. By decentralizing data storage and implementing robust encryption mechanisms, the platform addresses key concerns related to data privacy and integrity in healthcare systems. The secure exchange of information between patients and doctors facilitates efficient diagnosis and treatment, ultimately improving patient outcomes. However, continuous monitoring and updates are necessary to adapt to evolving security threats and regulatory requirements in the healthcare sector.

Continuous Security Enhancements: Regularly update encryption algorithms and security protocols to stay ahead of emerging threats and vulnerabilities.

User Education: Provide comprehensive training and education to users (patients, doctors, administrators) on best practices for securely accessing and managing electronic healthcare records.

Scalability Planning: Anticipate future growth and scalability requirements of the platform to accommodate an increasing volume of healthcare data and users.

8. Future Enhancements

In the future, enhancing the above project could involve several pivotal advancements. Integrating smart contracts into the blockchain architecture would automate and enforce access control processes, streamlining verification and approval mechanisms for accessing electronic healthcare records. Augmenting identity management solutions, such as decentralized identifiers and verifiable credentials, would fortify authentication protocols, ensuring secure verification of users' identities while preserving privacy. Embracing interoperability standards like HL7 FHIR would enable seamless data exchange with external systems, fostering collaboration among healthcare providers and facilitating comprehensive patient care. Moreover, exploring advanced encryption techniques such as homomorphic encryption or zero-knowledge proofs could bolster data security by enabling computation on encrypted data without compromising confidentiality. Enhancing the immutable audit trail with additional metadata and context information would provide deeper insights into data access patterns and compliance adherence, ensuring accountability and transparency. Integrating with emerging technologies like AI and IoT could enrich healthcare data with real-time insights, enabling predictive analytics and personalized interventions for improved patient outcomes. Lastly, refining the user experience through optimized interfaces and workflows would enhance usability, encouraging broader adoption among patients, doctors, and administrators, and ultimately advancing the efficacy of healthcare information systems.

9. References / bibliography

[1] Rui Lebre, Carlos Costa, "An Efficient and Reliable Architecture for Distributing Medical Imaging Data", *2021 International Conference on e-Health and Bioengineering (EHB)*, pp.1-4, 2021.

[2] K. Swetha, C.Ishaq Shareef, G. Sreenivasulu, K. K. Baseer, M. Jahir Pasha, "Study on Implementation of Electronic Health Records using Blockchain Technology", *2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pp.607-611, 2023.

[3] K. K. Baseer, B. Jaya Naga Varma, B. Harish, E. Sravani, K. Yashvanth Kumar, K. Varshitha, "Design and Implementation of Electronic Health Records using Ethereum Blockchain", *2023 Second International Conference on Electronics and Renewable Systems (ICEARS)*, pp.784-791, 2023.

[4] Caspar Von Lengerke, Alexander Hefele, Juan A. Cabrera, Oliver Kosut, Martin Reisslein, Frank H. P. Fitzek, "Identification Codes: A Topical Review With Design Guidelines for Practical Systems", *IEEE Access*, vol.11, pp.14961-14982, 2023.

[5] Narendra Kumar Rout, Debabrata Dansana, Nirjharinee Parida, Ranjeet Kumar Rout, "Improving Performance of Electronic Healthcare Record Management Systems (EHRMS) using Low Complexity Blockchain", *2022 Second International Conference on Computer Science, Engineering and Applications (ICCSEA)*, pp.1-5, 2022.