International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V71I4P208

Volume 71 Issue 4, 79-96, April 2023
© 2023 Seventh Sense Research Group®

Original Article

A Quantum Resistant Blockchain System for Privacy Protection of Patient Records

Smita Bansod[1], Lata Ragha[2]

[1]Shah & Anchor Engineering College and Research Scholar, Terna Engineering College, Maharashtra, India.
[2]Fr. C. Rodrigues Institute of Technology, Vashi, Mumbai University, Maharashtra, India.

[1]Corresponding Author : sakec.smitab@gmail.com

Received: 12 November 2022 Revised: 30 March 2023 Accepted: 13 April 2023 Published: 25 April 2023

Abstract *- In the present digital era, Personal Data Privacy is considered one of the fundamental rights in many countries, and regulation demands strict compliance with privacy laws. Patient Health Records in electronic form are highly personal and must be handled with due care and sensitivity to ensure the individual's privacy by giving full control to the patient by employing a self-sovereign model and, at the same time, protected from attacks by hackers. While blockchain technology, with its distributed ledger and immutability, promises to take care of the basic privacy and security requirements of personal digital data, there are several areas where improvements are needed in order to make this technology a robust, practical system. This paper proposes a system which introduces the privacy protection mechanisms to be applied to a blockchain-based patient records system for full privacy protection. The data is shared between different stakeholders in an encrypted format with a session key operational for a predetermined amount of time. The session keys are managed by private certificate authorization with quantum resistance NTRU algorithm. A comparative analysis of various asymmetric key cryptography algorithms indicates that Enhanced NTRU is superior in performance and provides the best security. The encrypted Electronic Health Record (EHR) is stored using Interplanetary File System (IPFS) protocol, and its hashes are recorded on the Ethereum blockchain test network. IPFS solves the issue of storing a large amount of data on the blockchain, and encryption solves the data transparency with Public Key Infrastructure (PKI) to resolve the authentication of the stakeholders. The proposed system's response time, latency, resource utilization and efficiency have been assessed experimentally for various transactions. The proposed system ensures patient confidentiality while sharing health records, making it future-proof.*

Keywords *- Blockchain, Security, Privacy, NTRU, PKI, EHR, Ethereum.*

1. Introduction

Privacy preservation is a mandatory legal requirement in order to provide protection for individual data. Statistics reveal that over 70% of the world's countries have legislation to ensure data protection along the lines of the GDPR of the European Union. Serious research is in progress to preserve digital information privacy, using the help of advanced Cryptography, Artificial Intelligence and Blockchain technologies. Telephone numbers, contact details, credit card account data and customer addresses are some examples of personal data. When a person shares personal data with an authorized entity (human/device), then privacy has to be preserved by that entity. Other data from different domains like EHR (medical or personal data), Intelligence–Surveillance- Reconnaissance ISR (enemy threats or captured sensor data), Financial data (account balance, password, credit card, etc.) and Insurance data can be protected using the proposed system.

Electronic Health Record (EHR) is a combination of personal information (address, contact number), electronic medical records (past medical profile, clinical status, diagnosis, prescribed course of treatment and details of surgery) and patient health record (maintained and observed by the patient using personal assistant as a thermometer, Blood pressure or glucose monitoring machines). The EHR data [1] is quite valuable to pharmaceutical companies, healthcare organisations and the medical fraternity. Any violation of privacy requirements may seriously affect the patient's health recovery and finances. According to IBM, stolen healthcare data is the most valuable one as compared to other domains. In the year 2021, the healthcare industry encountered an average total cost of $9.23 million due to healthcare data breaches [2]. Healthcare data breaches have increased significantly over the years. Various types of threats are observed in EHR, and protection from these threats is crucial for patient privacy.

Most EHR data is centralised in hospitals and hence vulnerable to data breach attacks. Therefore, the prime need is to decentralize EHR to minimise attacks. Further, the distributed ledger architecture of Blockchain technology and other special characteristics make it attractive to be applied to EHR to boost the security of health records. Various researchers are working in the healthcare domain using blockchain technology with focused research on privacy preservation. The EHRs maintained and controlled by the patients themselves will not be affected by attacks on the centralised hospital database. High-profile persons are known to have incorporated applications in their mobile handsets to protect their data from cyber-attacks. Using such techniques, patient data should be secured. The EHR data should be under patient control through a self-sovereign system. When other stakeholders use the EHR, these data should be transferred securely, ensuring confidentiality and authenticity. To accomplish the security goals using blockchain technology, a system is proposed in this paper where data is encrypted with a secure NTRU algorithm along with a PKI mechanism for key exchange.

The self-sovereign patient healthcare system is developed using smart contracts on the Ethereum blockchain. The health record hash values are stored in on-chain health record databases of the Ethereum test network. The actual health record in encrypted format is stored using IPFS protocol, ensuring scalability and reliability while resolving the issue of secure blockchain large data storage.

Organization of the paper: Section 2 explains the Motivation and Contribution of the paper, followed by Section 3, which narrates related work. Section 4 mentions the proposed system, while Section 5 explains about different algorithms used in the implementation. The Proposed system's implementation is described in Section 6. Section 7 provides system results with a discussion.

2. Motivation and Contribution

All privacy preservation laws, like the GDPR, focus on enhancing the characteristics of security, privacy, and safety of digital transactions. The user will be required to take control of the transactions in order to ensure privacy. Some countries consider personal data privacy a fundamental right of the citizen. A literature survey indicates that blockchain safety, security and privacy can improve further.

Identity Theft Resource Center (ITRC) report [3] discovered that hacking reported the highest percentage of data breaches, namely 39 percent of incidents. 81% of non-sensitive records are vulnerable to hacking attacks. The most common data breach, namely unauthorized access, accounted for 36.5 per cent of the security violations. When personal data is used for digital transactions, the user's privacy is highly vulnerable to hacking. Protection of personal data on the ledger becomes the priority objective.

EHR involves a number of patients whose personal data and health-related information is available in digital format. This valuable database is highly prone to Ransom ware and Equifax attacks. When the hospital maintains EHR in a centralised ledger, there is always the risk of permanently losing the record due to inadvertent deletion by the handlers. Management of EHR by the service provider may not give the flexibility to the patient to access the record.

2.1. Problem Statement

Due to the high vulnerability of a centralized database system to data breaches and loss of records, the privacy and immutability of EHR data should be protected by providing a decentralized database with control of the EHR remaining with the patient.

2.2. Objective

- To design a patient-controlled self-sovereign Electronic Health Record system and enhance the privacy capabilities of such a system using effective protection algorithms.
- To evaluate, verify and validate the parameters of such a system by simulation.

3. Related work

Data privacy protection is a critical issue in the digital era due to the regulations enacted by various Governments. While Blockchain technology is generally considered quite effective in privacy protection [4], there are certain aspects where this technology does not conform with the data protection regulations [5]. Serious research [6] is being pursued in many countries to develop blockchain applications that comply with data protection regulations. One such blockchain architecture, which claims to be regulation compliant, uses the Hyperledger Fabric, as discussed in the paper [7]. Several Privacy-Preserving mechanisms for blockchain technology employing the self-sovereign identity concepts are mentioned in the paper [8]. Flexible access control of data processing is achieved using Homomorphic algorithms [9].

Further, Privacy Enhancing Techniques (PET), such as Zero Knowledge Proof (ZKP), Secure Multiparty Computation (SMPC), Ring Signature and Mixing, are used to support privacy protection in blockchain technology [10], [11]. Quantum-resistant algorithms are helpful in protecting the privacy of data from hackers who deploy Quantum Computers for attacks on digital data [12]. Post-quantum blockchain applications [13] are protected and made more secure using lattice-based algorithms.

Traditionally EHRs are centralized and stored in a common server. There are several advantages to decentralizing the EHR and making it easily accessible to patients through distributed servers at different locations.

However, such a distributed system has security and privacy challenges that various researchers are addressing. EHR big data privacy in the cloud is preserved using data encryption methods [14]. EHR applications are protected using emerging blockchain technology [15]. Blockchain allows storing transaction data on-chain with limited bytes in the form of hash values in Merkle tree's root value for validation. Hackers/attackers target the EHR (through ransomware and Equifax attacks) to extort money or annoy a renowned person. Various incidents of failure of EHR security are mentioned in the paper [16].

The Interplanetary File System (IPFS) and SWARM distributed file system store and share huge amounts of data/content with content addressable hash. Cloud computing-based IPFS is a distributed system which protects the contents using dynamic encryption techniques. IPFS[17] is a lightweight technique as it supports transport encryption instead of content encryption [13], ensuring end-to-end data security. However, the data can be viewed by anyone with the Content ID (CID), which can seriously compromise the privacy requirement. IPFS has the facility to choose the method to confirm the application's security. Data privacy is protected using IPFS by signature [34] or encryption mechanisms [19]. Blockchain with IPFS is a good solution to preserve data privacy with cryptographic techniques[20].

Confidentiality and authentication are achieved using cryptographic algorithms. Some of the most popular symmetric key cryptography algorithms are ASE and 3DES, while RSA and ECC are the popular asymmetric key cryptography algorithms. The immediate future throws up the need for quantum-resistant cryptographic algorithms. Some of these algorithms suggested by NIST employ cryptography techniques which are Lattice-based, Hash-based, Multivariate Quadratic-based, Code-based or Supersingular Isogeny-based [21]. In order to achieve high security, decent speed and hacking resistance, the NTRU [22] algorithm using lattice-based SVP is preferred. It is widely believed that the decryption capacity of the NTRU algorithm can be further improved.

The Symmetric and asymmetric cryptographic keys are part of the public/private key infrastructure and should be shared securely with the authentication system. Various advanced key distribution techniques are available, which ensure a high level of security. The key distribution done with Certificate Authority (CA) has the risk of dishonest or damaged CA. The existing PKI systems use X.509 certificates employing various certification approaches like log-based, Web of Trust or distributed peer-to-peer (blockchain-based) certification, as mentioned in [23]. Each one of these approaches has some limitations or the other. In the traditional PKI Certification approach, the revocation mechanism remains an issue yet to be resolved. Using the Needham Schroeder-like protocol, the pre-shared key [24] concept enhances the key distribution method. The keys are shared securely using Shamir's algorithm [25] for document management. The Blockchain-based PKI-related metadata stored in X.509 certificate extension field [26] is effective, but the size of the certificate remains an issue. Public key Infrastructure (PKI) is now implemented with the help of blockchain using smart contracts [35].

4. Proposed System

The Electronic Health Record of a patient should be handled only by the patient to ensure the protection of privacy. In the self-sovereign model [28], the user (patient) is providing the self-identity to access the services offered by the service provider (Doctor). The identities (keys) are verified through the system, as shown in Figure 1. Blockchain technology offers the risk-free use of self - transactions without the assistance of any third party.

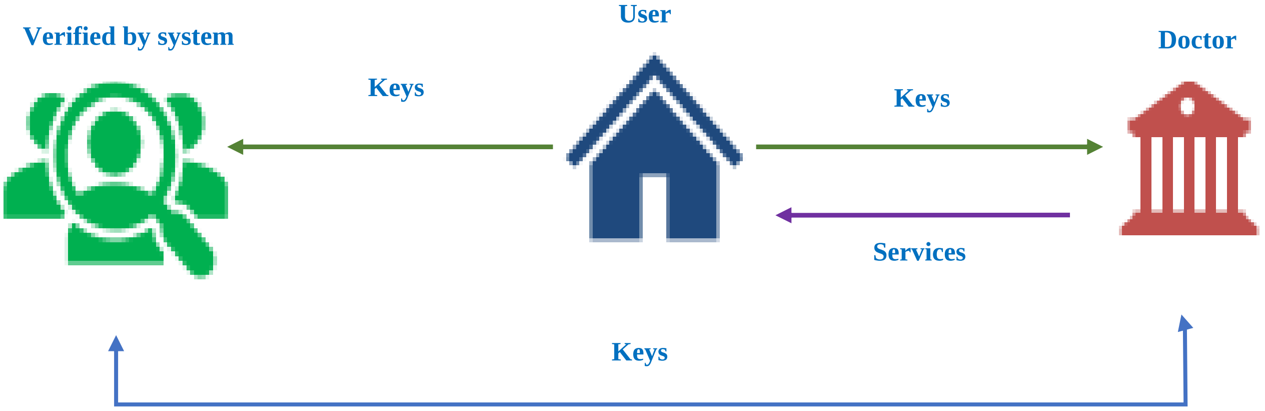


Fig. 1 Self-Sovereign System

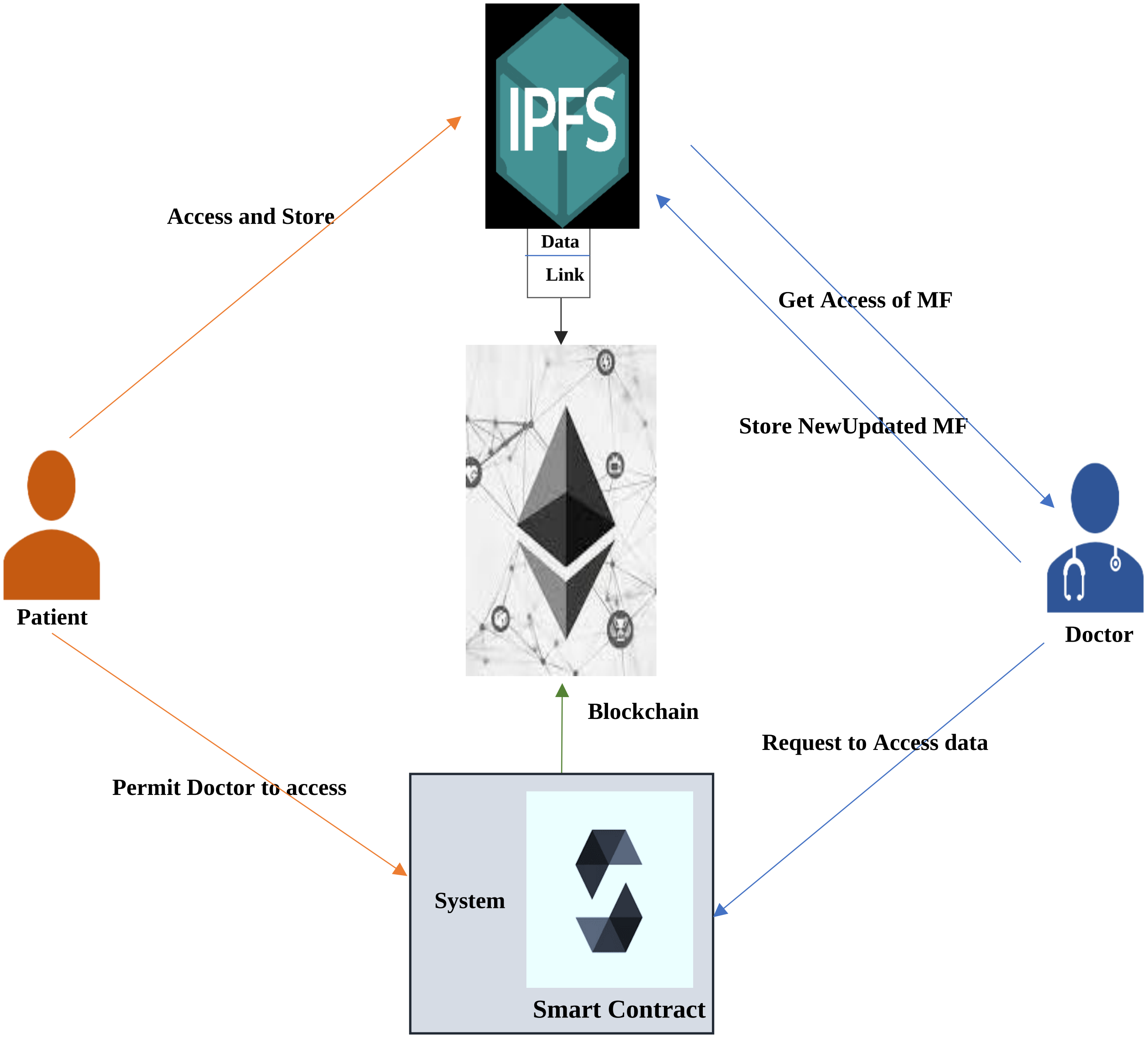


Fig. 2 Proposed System

EHR of patients can be designed either as a permissionless or a permissioned blockchain using a self-sovereign identification system. Both these systems have advantages and disadvantages but have the common objective of privacy protection. Privacy collapses due to transparency, as in the permissionless operation or to trusted third-party involvement under organizational control, as in the permissioned blockchain. Ethereum transparency issue can be resolved by adding encryption mechanisms with the support of PKI, and distributed architecture, eliminating the need for a trusted third party. The open-source Ethereum, reinforced with blockchain phase 3.0 and the Smart Contract, significantly increase the trust required for secure applications. The application can be developed using solidity language by incorporating algorithms or protocols as per requirement.

Healthcare data is voluminous in various formats and types such as PHR, EMR etc. Storing such huge data on the blockchain is an impossible task. So, there is a need to find a distributed database that will handle the auditability requirement of both data and transactions. As per literature survey, IPFS appears to be a suitable distributed file technique to store the data with blockchain taking care of the content address protection requirement. If the content location is leaked, data may get compromised, so there is a need to protect data privacy using strong cryptographic algorithms.

Advanced Encryption Standard (AES) algorithm, which uses symmetric key cryptography, effectively encrypts data in the form of images. The session key should be securely shared between the stakeholders using an asymmetric key cryptography algorithm. A literature survey indicates that the NTRU algorithm is a good choice as it is quite secure and free from patent restriction, making it available to anyone. The public keys which are needed to authenticate should be shared among the authorized stakeholders only with foolproof Public Key Infrastructure – PKI.

The proposed system, which applies a cryptographic process for enhancing security and privacy to patient data, is shown in Figure 2. Blockchain technology ensures the immutability of health records while storing data. The self-sovereign architecture ensures the control of the personal records by the patients, and any access can only be with permission from the patient. The IPFS database enables the storage of the encrypted data's hash values, so the health records are made tamper resistant. Efficient and secure access control is ensured through encrypted PKI between patients and doctors. The smart contract deployed on blockchain mentions the control process, access policies and authorities designated to update records.

The flow of the system is given in Figure 3. The system sequence diagram explains the patient's self-sovereign process:

- On request, the patient data is shared with the medical professionals.
- A session key S is generated for a particular session and stored in IPFS.
- The encrypted session key and the encrypted data file are sent to the doctor.
- The encrypted session key is also available to the patient.
- After decrypting the session and the file, the doctor updates the records and uploads them to the IPFS.
- A notification is sent to the patient after updating the record.
- The patient downloads and decrypts the record on the IPFS with the session key and his private key.
- Finally, the original record is modified based on the newly updated data by using the patient's public key, and the modified file is stored in the IPFS.
- The session key expires on completion of the session.

4.1. NTRU Algorithm

Future-proof cryptography drives us to think about quantum-resistant algorithms for all digital transactions. Out of the various quantum-resistant algorithms standardized by NIST, the Lattice-based NTRU algorithm [22] appears to be the one the industry prefers. NTRU is the acronym for 'Nth degree Truncated polynomial Ring Units', and the Institute of Electrical and Electronics Engineers (IEEE) standardized the NTRU Cryptosystem in the year 2009. As patent norms no longer apply to NTRU, researchers are digging more into the NTRU algorithm, intending to enrich it further. NTRU provides high security and speed with less complexity. The throughput and quantum-resistant security of NTRU are better as compared to RSA and ECC.

NTRU uses polynomial Z and variable X in the format of ring R [29] with an integer of maximum degree N which is truncated upto N-1 as R = Z[X]/[X^N-1). The various variables/notation used in the NTRU algorithm is given in Table 1.

Table 1. NTRU Parameters

Variables	Description
N	Polynomial ring dimension with maximum degree N
p	Small modulus
q	Big modulus
f	Private Key
h	Public Key
f'	Polynomial that is used to calculate the value f
g'	A provisional polynomial used in the key generation process
r'	Blinding polynomial use in encryption
R'	Random ring equation generation
h'	Polynomial of Public Key
fp'	Polynomial Multiplicative inverse of (f mod p)
fq'	Polynomial Multiplicative inverse of (f mod q)
M	Plaintext message
%	Modulus
T	Truncate
E'	Encryption polynomial

4.1.1. Key Generation

NTRU generates two keys, private and public keys, in polynomial form. f' as Private key f in polynomial form is calculated with XN-1 and various random variables. Public key h is derived as:

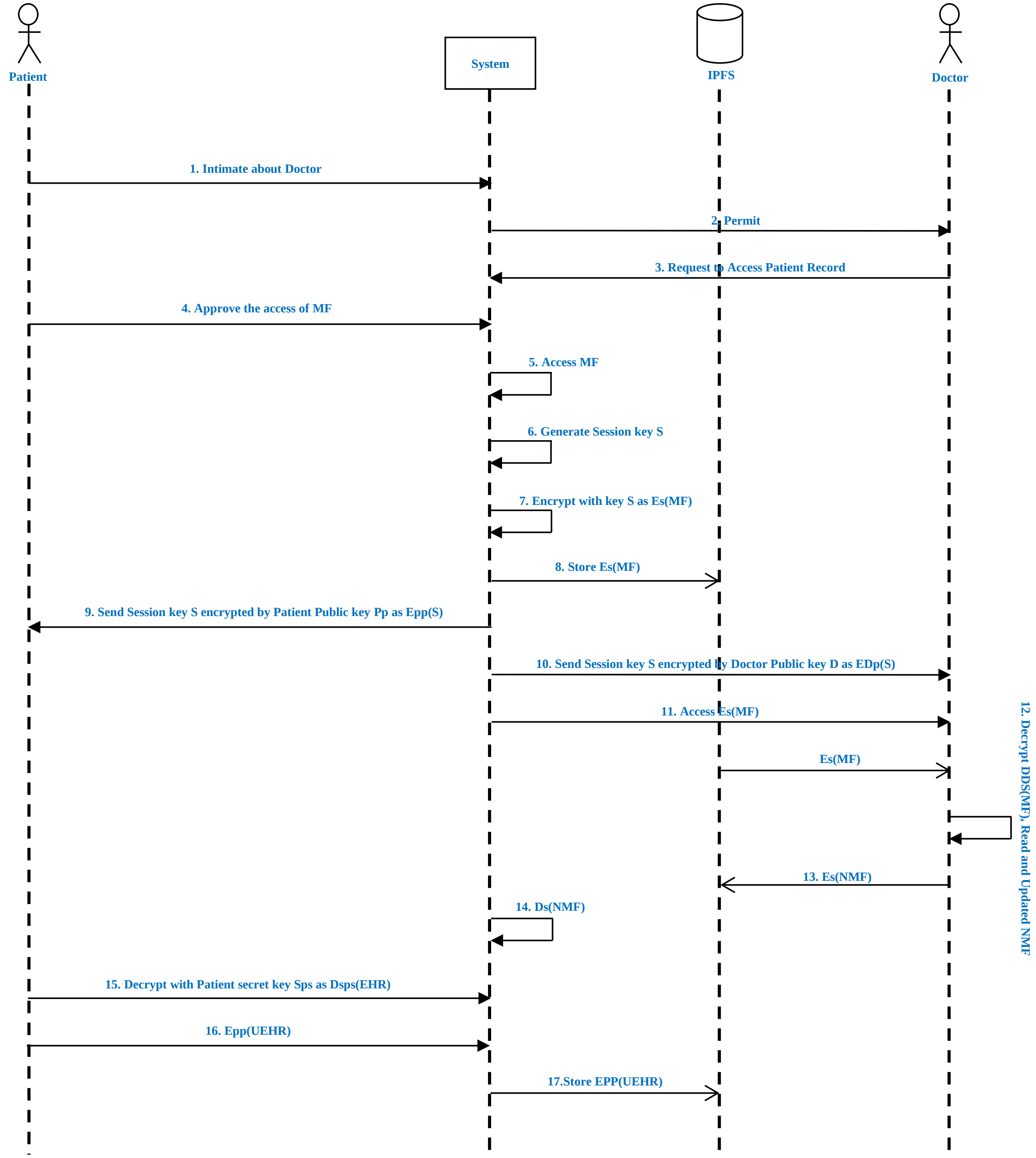

Patient
System
IPFS
Doctor
1. Intimate about Doctor
2. Permit
3. Request to Access Patient Record
4. Approve the access of MF
5. Access MF
6. Generate Session key S
7. Encrypt with key S as Es(MF)
8. Store Es(MF)
9. Send Session key S encrypted by Patient Public key Pp as Epp(S)
10. Send Session key S encrypted by Doctor Public key D as EDp(S)
11. Access Es(MF)
Es(MF)
12. Decrypt DDS(MF), Read and Updated NMF
13. Es(NMF)
14. Ds(NMF)
15. Decrypt with Patient secret key Sps as Dsps(EHR)
16. Epp(UEHR)
17.Store EPP(UEHR)

Fig. 3 Sequence diagram of the proposed system

Notations:

MF	– Medical Record File	Es	– Encryption with the Session key
S	– Session Key	Pp	– Public key of patient
E	– Encryption	Ps	– Secret key of patient
D	– Decryption	Dp	– Public key of doctor
		Ds	– Secret key of doctor

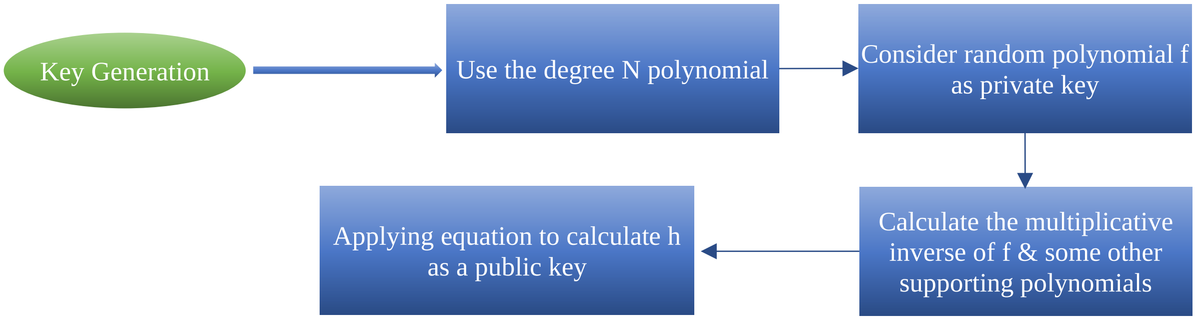

Key Generation
Use the degree N polynomial
Consider random polynomial f as private key
Calculate the multiplicative inverse of f & some other supporting polynomials
Applying equation to calculate h as a public key

Fig. 4 NTRU Key generation algorithm

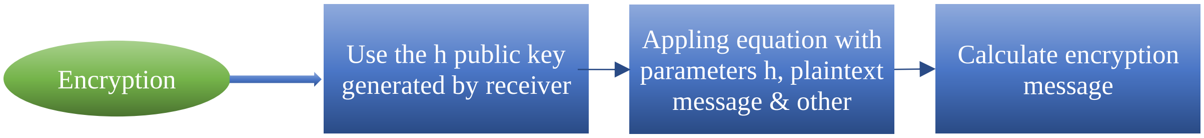

Encryption
Use the h public key generated by receiver
Appling equation with parameters h, plaintext message & other
Calculate encryption message

Fig. 5 NTRU encryption algorithm

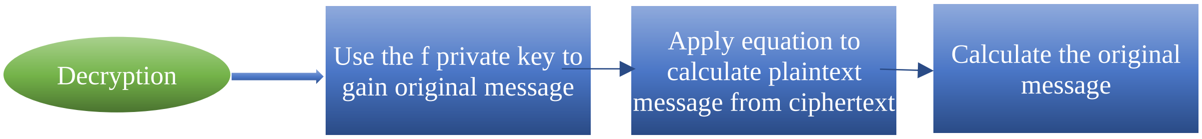

Decryption
Use the f private key to gain original message
Apply equation to calculate plaintext message from ciphertext
Calculate the original message

Fig. 6 NTRU decryption algorithm

$$pfq' = (p*fq').\ T(q) \qquad (1)$$

$$h = [((pfq').T(q))\ \%\ r'].\ T(q) \qquad (2)$$

2. Encryption: Plaintext P as polynomial converted into ciphertext after getting h as the public key of the receiver, and encryption takes place as::

$$E' = [((R'*h).T(q) + M)\ \%\ R'].T(q) \qquad (3)$$

3. *Decryption:* The receiver uses his private key to decipher the message as:

$$a' = (f' * E')\ \%\ R').\ T(q) \qquad (4)$$

$$b' = (a'\ .T(q)) \qquad (5)$$

The original message appears after decryption without errors by carefully selecting the q value between - q/2 and + q/2 such that T(p) and T(q) are appropriately connected.

The message is encrypted with the help of a symmetric key cryptographic algorithm named Advanced Encryption Standard AES [30]. The session/shared key of AES, which is randomly generated, should be distributed securely using public/asymmetric key cryptography. There is a need for Public Key Infrastructure (PKI) to make the secured and authenticated key distribution. The private keys are with the individual stakeholders, while the corresponding public keys are stored in Ethereum. The proposed system adopts a new strategy to do away with the requirement for a trusted CA to issue the certificates, simplifying the end users' wallet management functions.

The steps for a secured key management system embedded in our proposed method for providing authentications of the stakeholders:

- The patient requests the public key of Doctor Dp from the Ethereum Blockchain.
- The Doctor's public key to the patient is returned via the Ethereum Blockchain.
- The patient encrypts a Nonce (N1) that uniquely identifies a transaction and patient's identity (IDA) with Doctor Dp's public.
- The Doctor requests Patient Pp's public key from the Ethereum Blockchain.
- The Ethereum Blockchain responds to the doctor's request with the patient's public key.
- The Doctor delivers the already received Nonce (N1) to the patient, as well as a second Nonce (N2) encrypted with patient Pp's public.
- The patient re-sends the encrypted Nonce (N2) to the doctor.
- The entire communication is encrypted using the public key of Doctor Dp, and a created session key S is signed with the patient's private key.
- The Doctor decrypts the message and confirms the signature using his private and patient Pp's public keys.

The patient and the doctor can access the session key S. This key may be used to establish any secure connection between the server and the end devices.

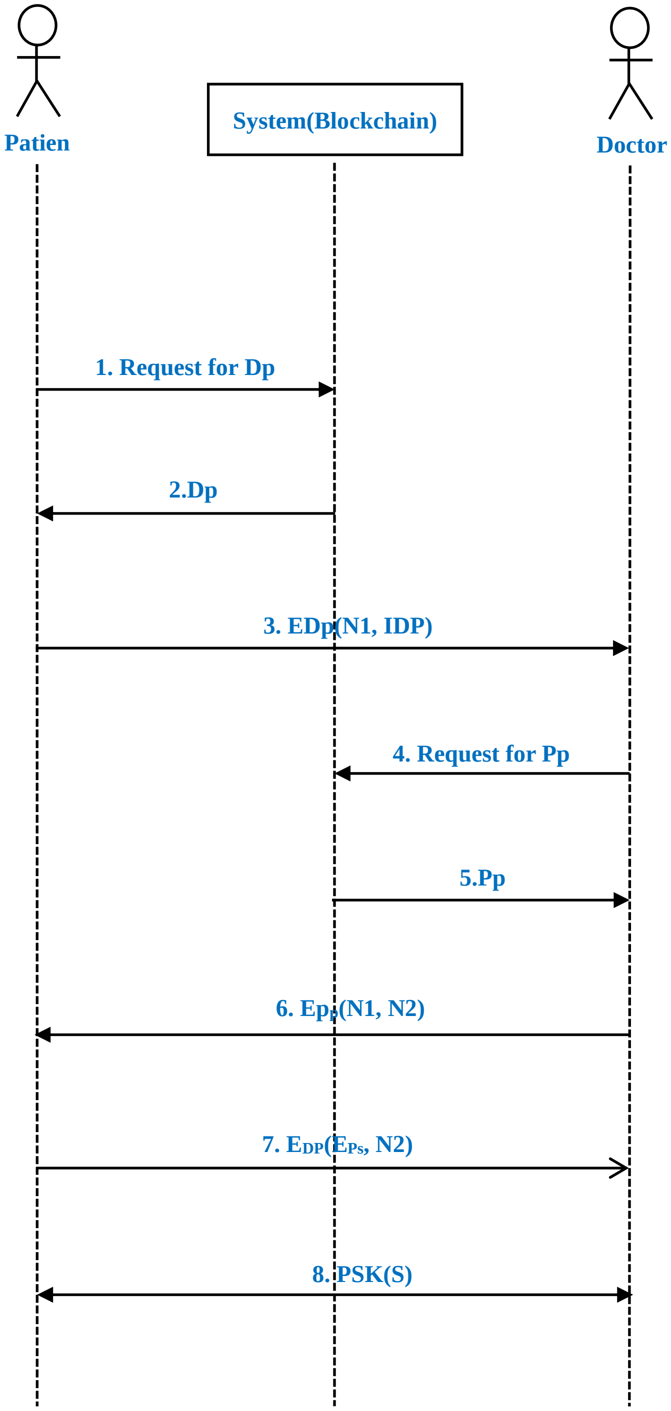


Fig. 7 Key exchange protocol

Notations:
E – Encryption
D – Decryption
S –Session key

Pp – Public key of patient
Ps – Secret key of patient
Dp – Public key of doctor
Ds – Secret key of doctor

5. Algorithm Design

A system is proposed here to provide data security and preserve the privacy of patient EHR. Various functions will be discussed to fulfill the requirement of patient privacy using blockchain technology.

While accessing the application, to ensure the privacy of the user (patient or doctor), a registration process becomes necessary. Here the user fills in the needed information with the user's details, and user data is stored on the blockchain. At the same time, the blockchain network node is created through the smart contract for that user.

```
Algorithm 1: Register_User
Purpose: Registering new users on the system
Input: User requests for registration
Output: User registered on the system and transaction done to generate user registration details in the block of blockchain through the smart contract
1. Enter the details in the registration form by the user
2. Select role P(patient) or D(doctor)
3. If Ui ← Add the details
4.   Then Metamask popup for transaction confirmation as per the smart contract
5.         Ti ← confirm
6.         Ui login was generated, and Node Ui on the blockchain network
7. Else popup to add all details
```

The key exchange protocol shown in Figure 7 governs the Key Exchange operation. After registration, the patient and the doctor confirm their identity to each other and exchange session keys, as explained in Algorithm 2 below.

```
Algorithm 2: Key_exchange()
Purpose: Sharing session key S using NTRU public key cryptographic algorithm for EHR encryption/decryption.
Input: Two users say P(patient) and D(Doctor) are registered and login ID generated.
Output: S exchanged between P and D
1. P login {
2.   P find the identity of D from the blockchain registration block.
3.   If D ← Registered
4.   Then P gets Dpu (Doctor's Public Key)
5.         P generate N1 (Nonce/challenge)
6.         P send N1 and P identity
7.         Smart contract executes the transaction and generates block }
8. D login {
9.     D check accept the challenge
10.    D view N1 and P identity
11.    D generate N2
12.    D send N2 and N1 and P Identity
```

```
    Smart contract executes the transaction and generates block }
P login {
    P Check and view challenge N2
    P exchange D identity and N1 and N2
    Smart contract executes the transaction and generates block }
D login {
    D view final challenge
    Respond to view challenge
    Smart contract executes the transaction and generates block }
P and D confirm the identity
P generate random S and shares with D public key
```

In order to keep the file secure, the file is uploaded on the blockchain in an encrypted format, and the hash value is stored on IPFS. Sharing files between Patient and Doctor can occur by Upload_file function using the AES algorithm and session key sharing using the Key_exchange function.

```
Algorithm 3: Upload_file()
Purpose: Uploading a file on the blockchain
Input: S Key should be exchanged
Output: Encrypted file uploaded on IPFS blockchain
F ←the file which wants to upload
Es(F) by AES (Es encryption by S)
Choose F
Call goes to Metamask to execute the smart contract to upload the file on the blockchain
HashF generated
```

Patients or Doctors can download the file on the application without downloading on to the local system from the blockchain and verify with HashF to achieve integrity.

```
Algorithm 4: Download_file()
Purpose: Downloading files from the blockchain as per request of the user
Input: Get the list of the downloaded file
Output: The file is visible in the user application system but not saved on the local system
User check download list option
Checklist
Fi, which wants to download
Call goes to Metamask to execute the smart contract to download the file from the blockchain
Ds(Fi) by AES (Ds decryption by S)
F visible in the user application
```

If the patient wants to share the file with the doctor or vice versa, the uploaded file by the patient is in an encrypted format. As per requirement, the file will be shared where the session key is already shared by the Key_exchange function to other users and decrypted using S by the download_file function.

```
Algorithm 5: Share_file()
Purpose: Sharing the file to the respective user account
Input: File should be uploaded on the blockchain which the user wants to share, and other users should be registered
Output: File is available in the user-shared list with whom it was shared
User 1 call upload_file(Fi)
Available in user 1 sharable list
Share file Fi with user 2
Call goes to Metamask to execute the smart contract to share the file from the blockchain
User 2 login {
    Check download list
    Fi is available in the download list
    User 2 call download_file(Fi)
    Fi visible in User 2 application }
```

The patient wants to keep a detailed patient history, and the records are created. Also, PHR and EMR can be recorded for sharing with doctors after converting into the file.

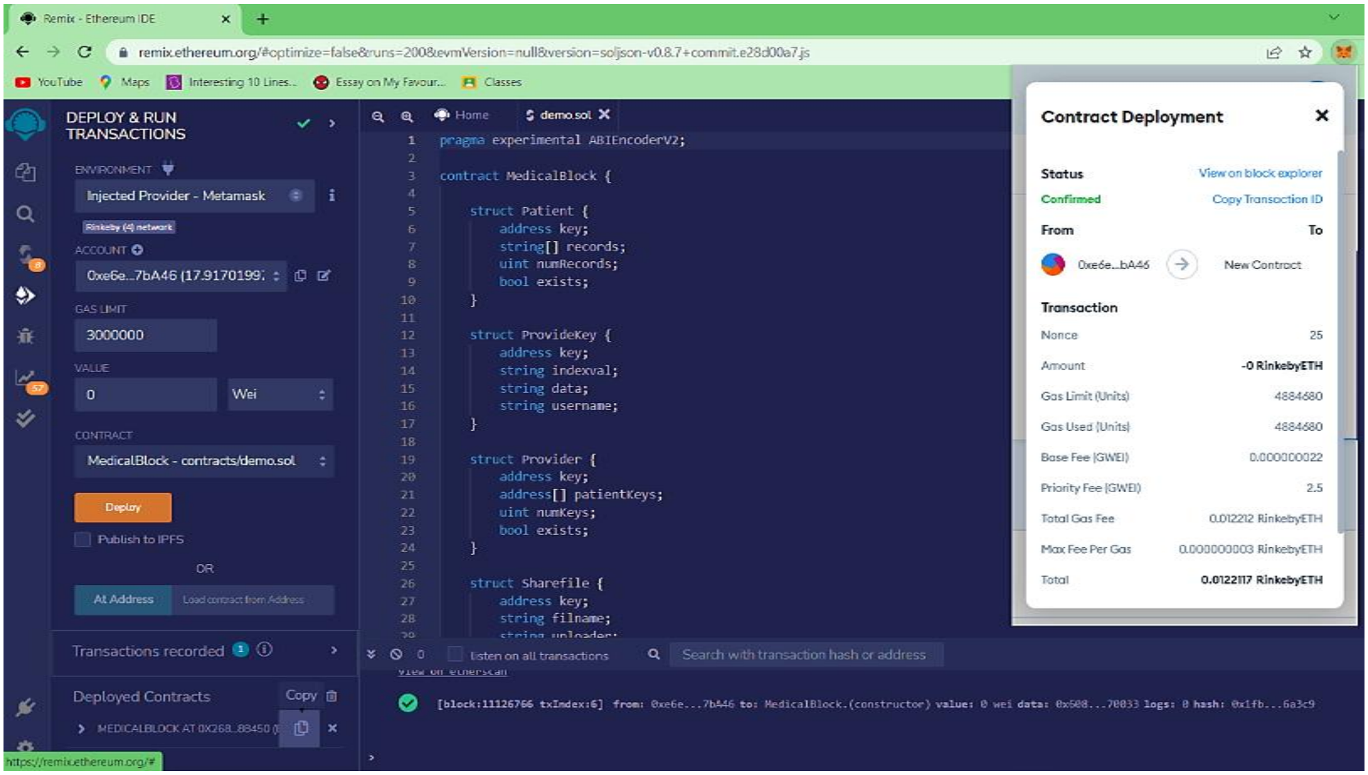
```
Algorithm 6: Add_record()
Purpose: Storing the EHR or other information in record format
Input: Other information like PHR and EMR are kept ready to create a record
Output: PHR, EMR, and EHR can be viewed whenever the user wants to check the records
Create a series of records
Add information
View whenever required
```

6. Implementation and Results

The proposed system is implemented with the help of a test network to simulate the behavior of the Ethereum main network. The smart contract is written in solidity using remix IDE employing Metamask and Windows 10. While writing the contract, care is taken to avoid the issues of attacks [31].

NTRU algorithm is implemented with Numpy, SymPy Python library for symbolic mathematics. fp polynomial is calculated with the help of an inverted polynomial function passing f', R' and p parameter values. Decryption error can be minimized with the help of a truncated function (Trunc/T) by truncating each equation with q values between - q/2 and + q/2. Decryption occurs with the help of p truncate.

The system starts functioning on the deployment of smart contracts on Ethereum IDE. The transaction gets confirmed with the connection of the Metamask wallet, and the contract starts working, as shown in Figure 8.


Remix - Ethereum IDE
remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.7+commit.e28d00a7.js
YouTube
Maps
Interesting 10 Lines...
Essay on My Favour...
Classes
DEPLOY & RUN TRANSACTIONS
Home
demo.sol
Deploy
At Address
Load contract from Address
Copy
https://remix.ethereum.org/#

ENVIRONMENT
Injected Provider - Metamask
Rinkeby (4) network

ACCOUNT
0xe6e...7bA46 (17.9170199

GAS LIMIT
3000000

VALUE
0 Wei

CONTRACT
MedicalBlock - contracts/demo.sol

□ Publish to IPFS

OR

Transactions recorded 1

Deployed Contracts
MEDICALBLOCK AT 0X268...88450

```
pragma experimental ABIEncoderV2;

contract MedicalBlock {

    struct Patient {
        address key;
        string[] records;
        uint numRecords;
        bool exists;
    }

    struct ProvideKey {
        address key;
        string indexval;
        string data;
        string username;
    }

    struct Provider {
        address key;
        address[] patientKeys;
        uint numKeys;
        bool exists;
    }

    struct Sharefile {
        address key;
        string filname;
        string uploader;
```

□ listen on all transactions
Search with transaction hash or address

```
view on etherscan
[block:11126766 txIndex:6] from: 0xe6e...7bA46 to: MedicalBlock.(constructor) value: 0 wei data: 0x608...70033 logs: 0 hash: 0x1fb...6a3c9
```

Contract Deployment

Status: Confirmed
View on block explorer
Copy Transaction ID

From: 0xe6e...bA46
To: New Contract

Transaction

Nonce	25
Amount	-0 RinkebyETH
Gas Limit (Units)	4884680
Gas Used (Units)	4884680
Base Fee (GWEI)	0.000000022
Priority Fee (GWEI)	2.5
Total Gas Fee	0.012212 RinkebyETH
Max Fee Per Gas	0.000000003 RinkebyETH
Total	0.0122117 RinkebyETH

Fig. 8 Smart contract deployment

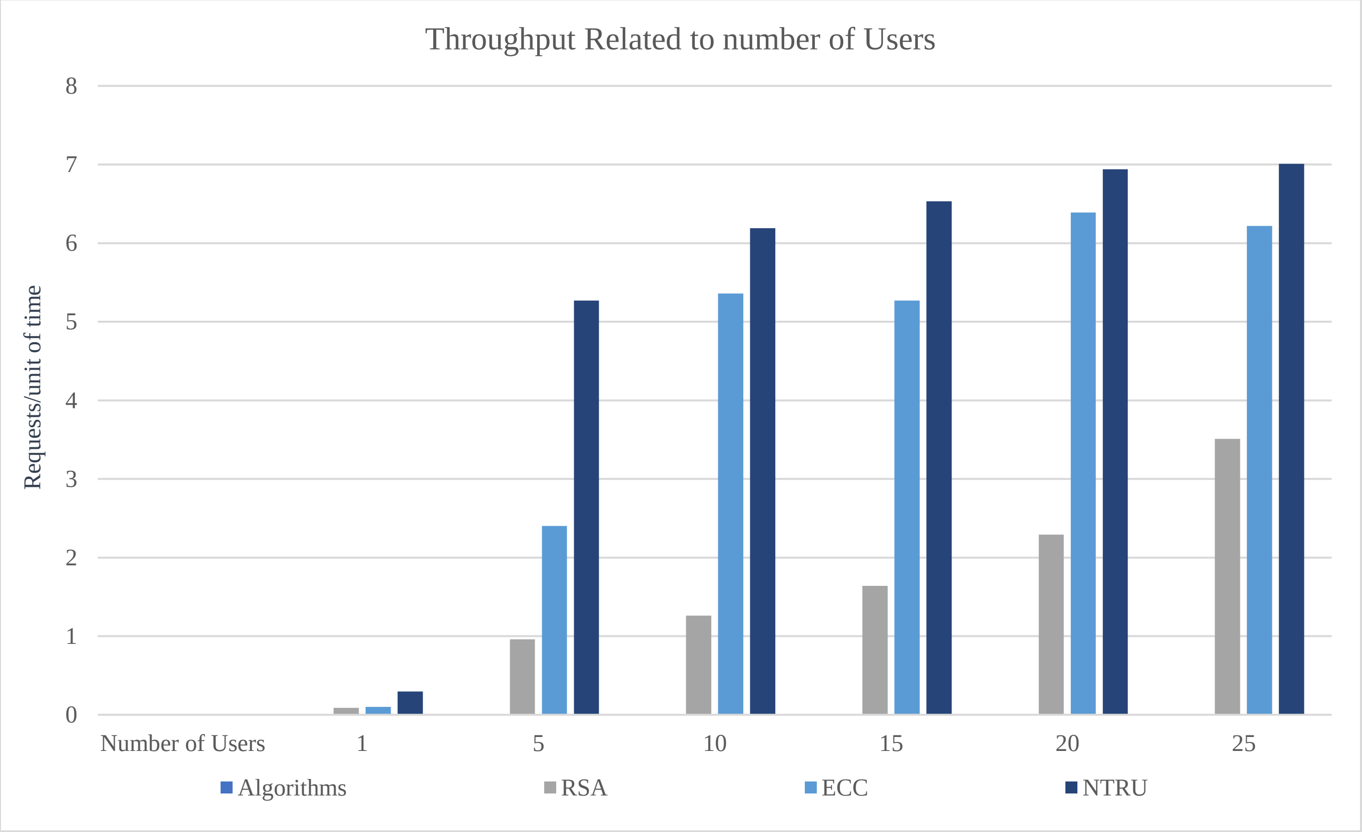


Fig. 9 Throughput comparison for various algorithms

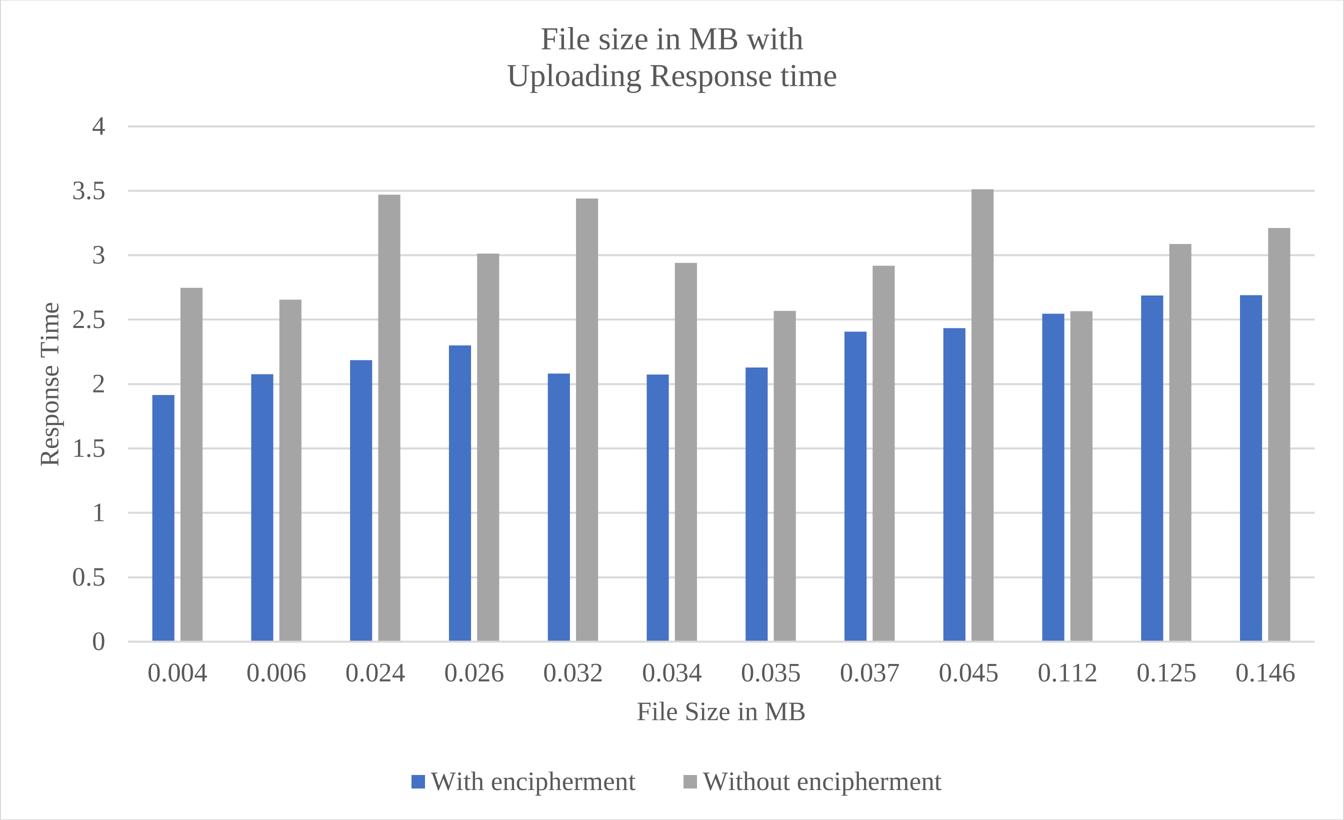


Fig. 10 Uploading response time

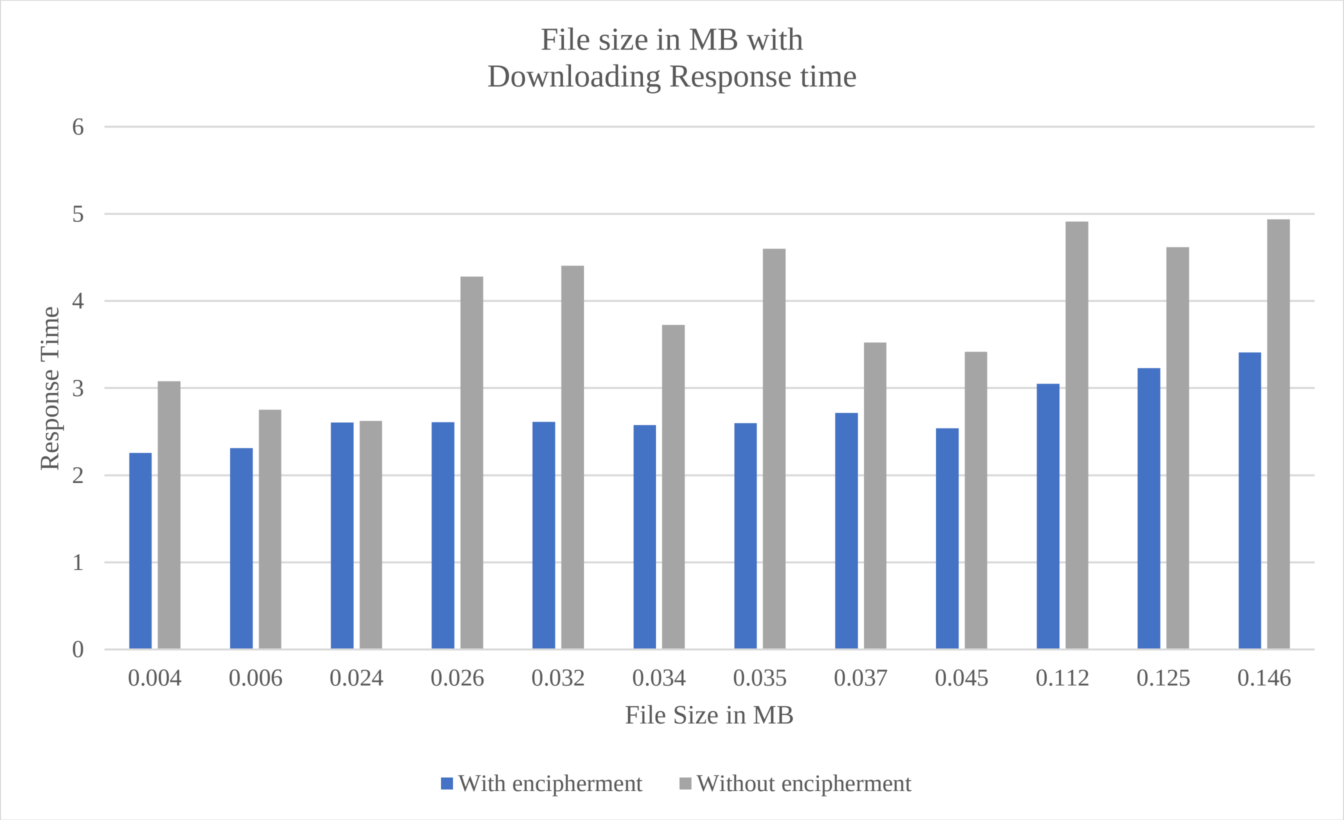


Fig. 11 Downloading response time

6.1. Throughput

The public key should be exchanged securely in addition to encrypting the file to maintain confidentiality. It has been reported [29] that the NTRU algorithm's performance is superior to RSA and ECC. The throughput can be represented in requests per second (minute/hour), and JMeter is used to evaluate the throughput under heavy load. The time unit is chosen, keeping 1.0 as a reference. The quantity of data downloaded from the server during the execution of the performance test can be seen on the JMeter as throughput. It may be seen from Figure 9 that the throughput related to the number of users is highest in the case of the NTRU algorithm as compared to RSA and ECC. Further, NTRU is suitable for key exchange and is considered quantum resistant.

6.2. Response Time

The time elapsed from when the mouse is clicked for uploading/downloading to the time it takes place (on the IPFS blockchain) is called response time. Here the uploading and downloading response times are calculated and analysed. The proposed system combines an encryption algorithm and NTRU based PKI system. The response time is compared for the two cases – (i) EHR file without encryption and (ii) with encryption. The uploading response time is shown in Fig. 10, and downloading response time is shown in Fig. 11, with and without encryption.

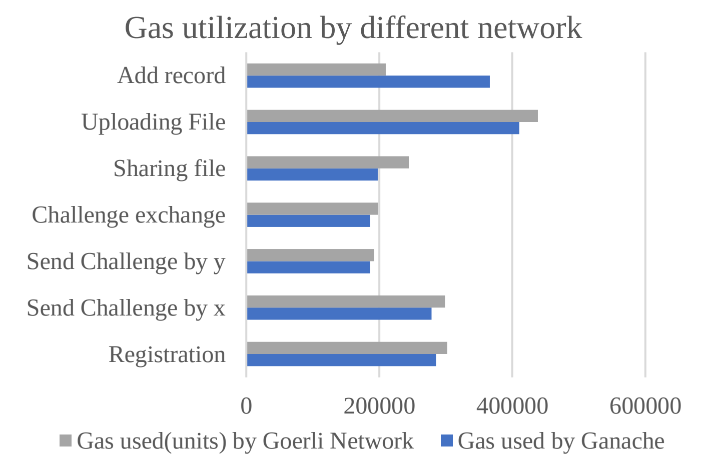
When the file is uploaded or downloaded on the blockchain in the direct form (text or image), the response time is higher than the response time if it is in the encrypted format because the encrypted file is compressed and in binary format.

6.3. Cost Projection

Transactions are based on the gas units required in the Ethereum network, and gas values are calculated from gas units. A practical system must perform various functions using gas values to access the blockchain. These are called transactions. The transactions are taking place on the Rinkeby network. The smart contract address is connected to the Rinkeby wallet address to utilize transaction gas value from the wallet account to execute the transaction through the contract. Cost projection is made with gas units for each function with the corresponding gas fees in Rinkeby Ethers (R-ETH). It may be seen from Table II that the total gas consumption to execute the contract is 6986676 units. Therefore, the relevant cost of contract deployment is 0.017238 R-ETH, with a corresponding price of $ 34.50. An exercise has been done to compare the gas values arrived at using the Goerli testnet for the proposed system with those arrived at using the Ganache network. It may be seen from fig. 12 that the Goerli test network utilizes slightly more gas units than the Ganache network.

Table 2. Cost Projection

Functions	Gas used by Ganache	Gas used(units)	Total Gas fees (R-ETH)	Total gas (R-ETH)
Contract deployment	4884680	4884680	0.012212	0.0122117
Registration	285348	302248	0.000733	0.00073295
Login	191233	205333	0.000451	0.00045133
Send challenge by x	278589	298689	0.000618	0.00061772
Send challenge by y	186197	197897	0.000438	0.00043799
Challenge exchange	186269	298689	0.000618	0.00061772
Sharing file	197661	391638	0.000824	0.00082397
Uploading File	410361	438261	0.001031	0.0010309
Add record	366338	209445	0.000314	0.00031417
Total	6986676	R-ETH: RinkebyETH		0.017238



Functions	Gas used by Ganache	Gas used (units) by Goerli Network
Registration	285348	302248
Send Challenge by x	278589	298689
Send Challenge by y	186197	192189
Challenge exchange	186269	197897
Sharing file	197661	244276
Uploading File	410361	438285
Add record	366338	209577

Fig. 12 Throughput comparison for various algorithms

6.4. Response time and latency for functions

Response time RT is the duration from the start time to the end time of the function to complete the transaction to validate and create the block on the blockchain. TS is the transaction start time of the function where transaction confirmation starts, and TE is the transaction end time where the transaction is completed to create a block of that transaction. Hence RT = TE - TS. Figure 13 shows the response time for the various function in the sequential order flow. Figure 14 shows the plot of the functions according to increasing response time.

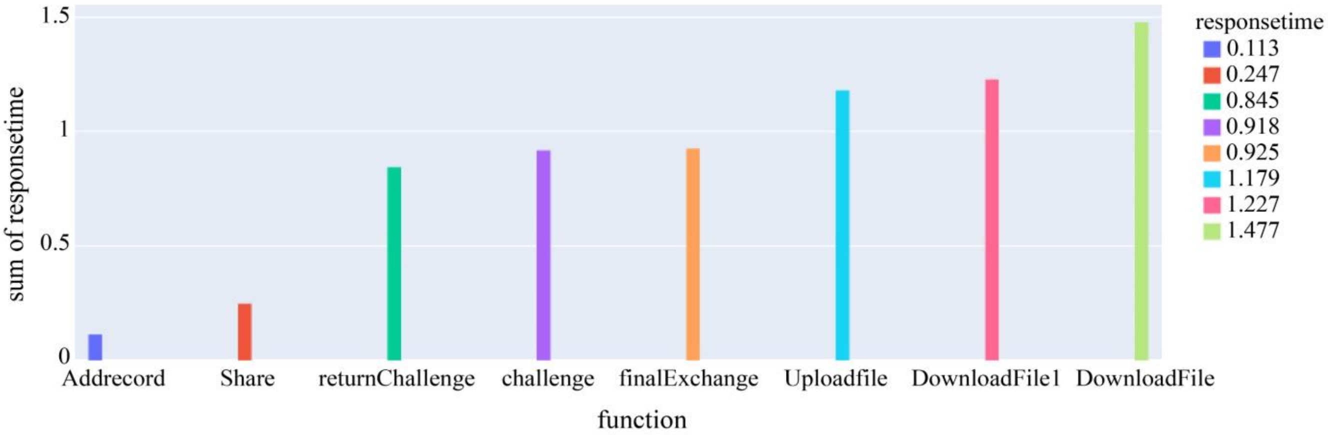


Fig. 13 Response time for various functions in sequence

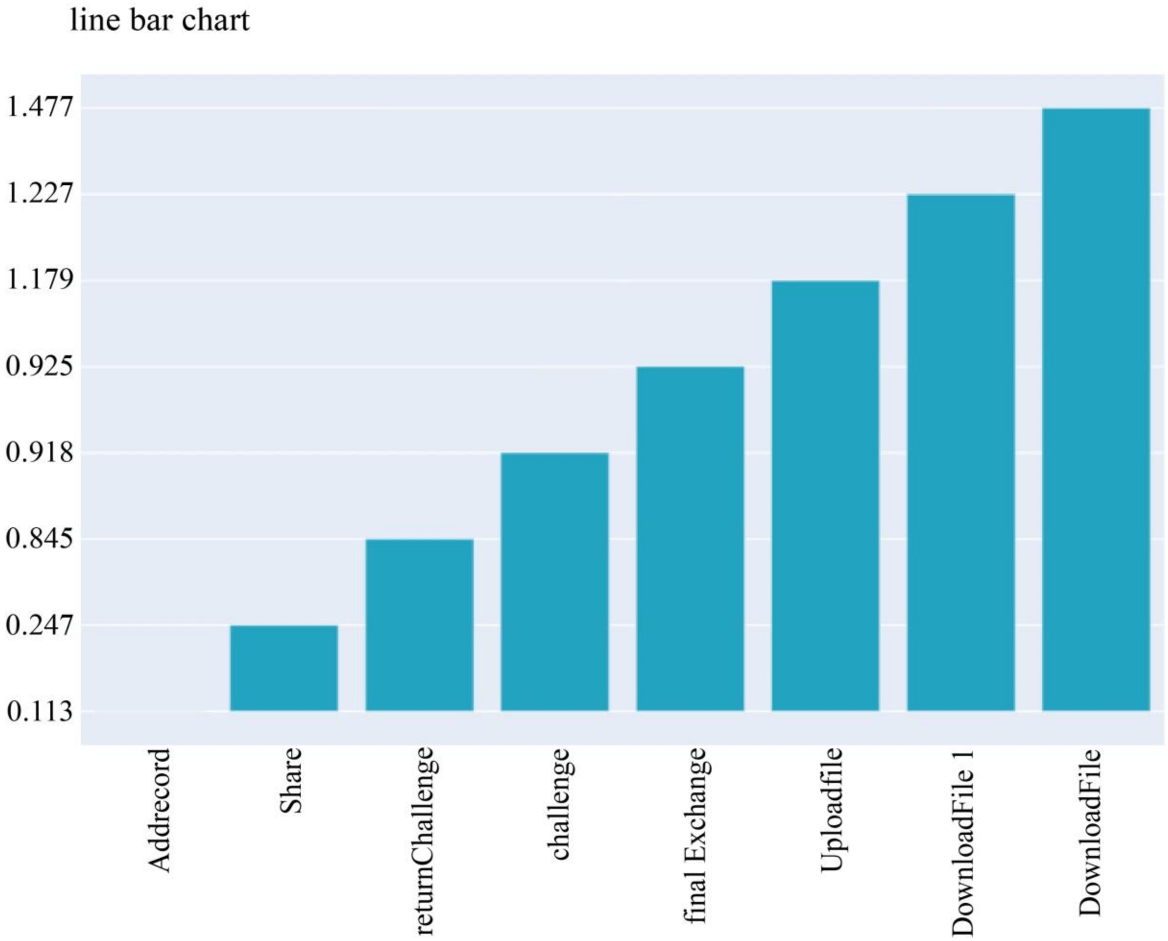


Fig. 14 Response time for various functions

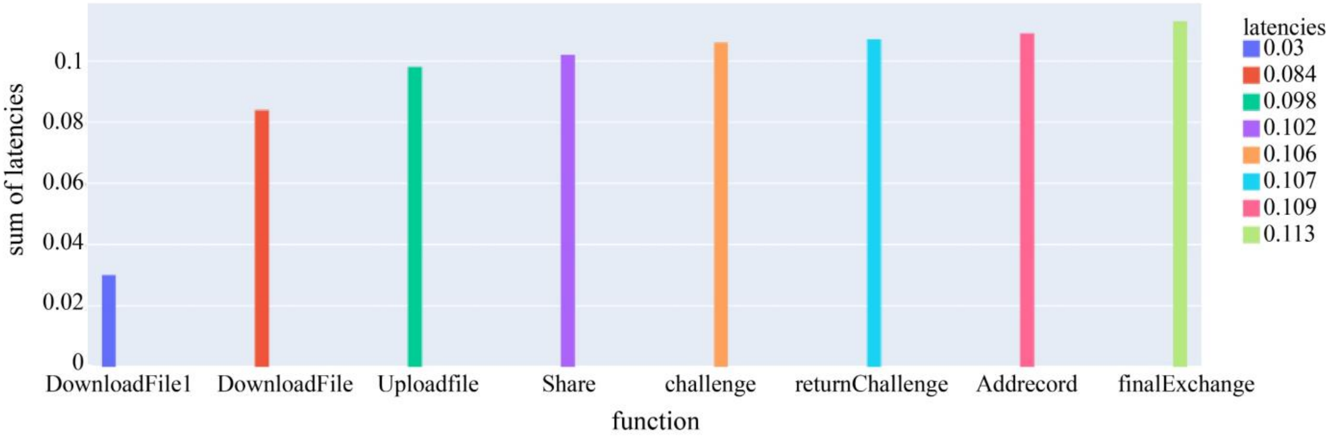


Fig. 15 Latency for various functions in sequential order

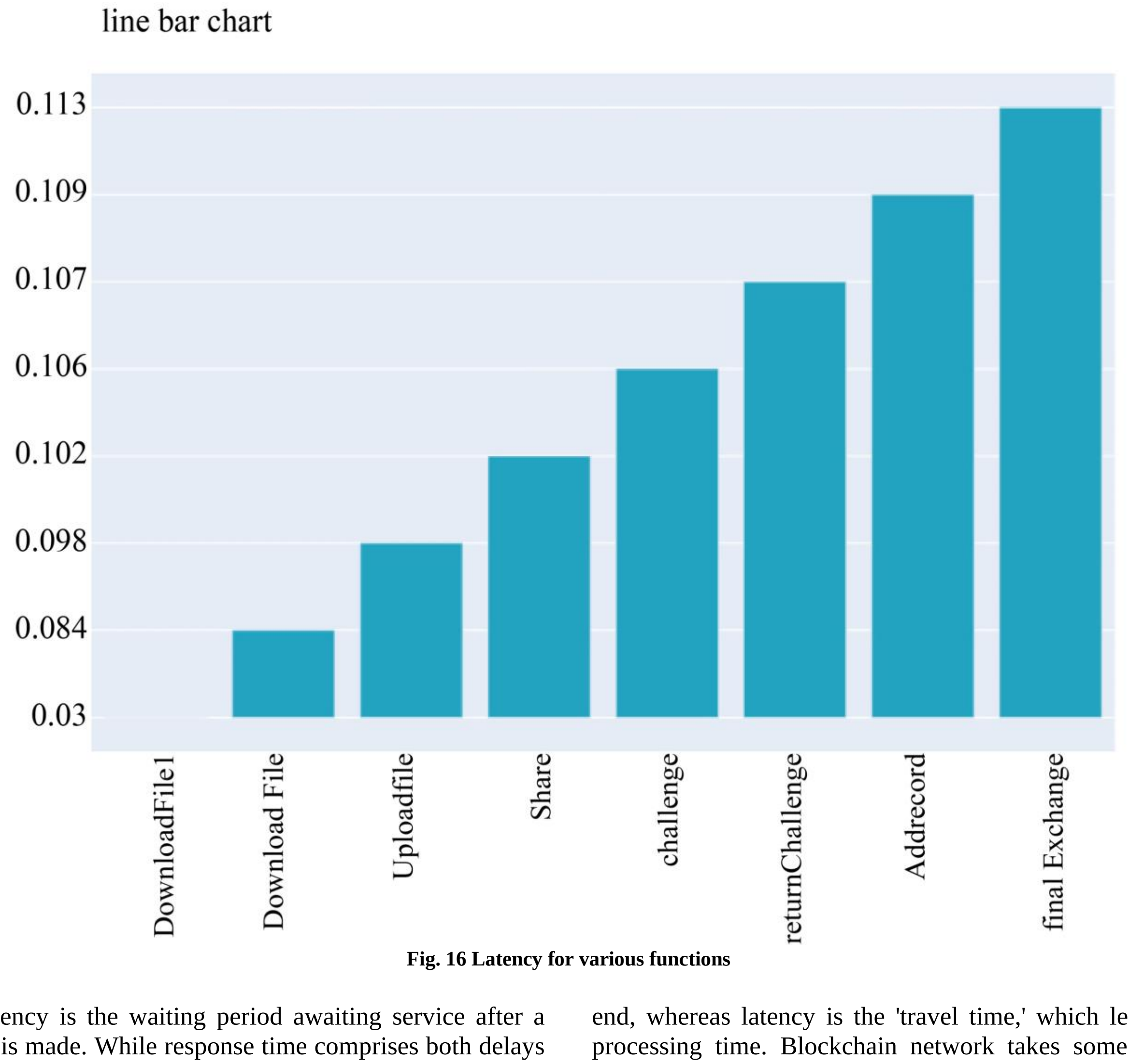


Fig. 16 Latency for various functions

Latency is the waiting period awaiting service after a request is made. While response time comprises both delays and real processing time, latency refers to the system's delay. Response time generally measures a network transaction from a client application's point of view from beginning to end, whereas latency is the 'travel time,' which leaves out processing time. Blockchain network takes some waiting time due to block validation. Figure 15 shows the latency of execution of different functions in sequential order, and Figure 16 shows functions according to increased latency.

7. Performance Analysis

7.1. Privacy Protection

The proposed system uses blockchain technology along with the cryptography algorithm NTRU for key exchange and the AES algorithm for information encipherment. Information is stored on IPFS in an encrypted format to protect privacy without data tampering.

7.2. Confidentiality

Encipherment algorithms provide confidentiality. Doctors and patients can access the data that should be protected from various attackers. The NTRU algorithm is used for key exchange, and the AES algorithm uses data encryption and decryption.

7.3. MIM Security

Lattice-based NTRU algorithm is used in the proposed system. On NTRU, the attack consists of two consecutive stages [32]: the first stage is to reduce the initial rows of the public NTRU lattice basis LNTRU with the best-known lattice reduction scheme, and the second stage is to store partial key guesses in boxes dependent on the output of the first stage. The hacking attack will be successful only if two partial key guesses collating in the same box can be combined to recover the entire private key. The parameters used in the proposed system for NTRU protect from MIM attacks.

7.4. Quantum Resistance

The proposed system employs the quantum-resistant properties of the NTRU algorithm. Lattice-based cryptographic constructions are based on the presumed hardness of lattice problems, the most basic of which is the shortest vector problem (SVP). An arbitrary input is given to the lattice with the objective of getting the shortest nonzero output. NTRU relies on the complexity of factorizing certain polynomials, making it resistant to Shor’s algorithm. In order to provide a 128-bits post-quantum security level, NTRU demands 12881-bits keys. No attack has been reported against the NTRU algorithm [33].

7.5. Security Theorem

Theorem 1: Assume that an adversary hacks into the data server and gets access to the preserved files stored in the server. The data will be safely preserved because the adversary will not be in a position to view, edit or delete the preserved files. Further, the user will be alerted if the file is destroyed.

Proof: The uploaded data M is actually stored in the blockchain, whose immutability characteristic ensures that the confirmed blockchain transactions can neither be tampered with nor edited nor deleted. It is impossible to construct an extensive new main chain that will be required to modify data, so attempts to tamper with confirmed data will not be successful. Hackers cannot recover and view the contents of confirmed files Fi as these are stored in blockchain with irregular names with encrypted indexes on IPFS at random file locations, which are also encrypted. Any tampered file F’ will be exposed when its Hash value is compared with the HashF preserved in the blockchain.

Theorem 2: The user's sensitive data cannot be stolen by the adversary because of the enhanced encryption techniques employed in the system, which cannot be decrypted by the adversary even if the attempt to pinpoint all the blockchain transactions is successful.

Proof: With the user protecting the private key with the Ethereum wallet and a secure key exchange process being in place, there is no way anyone can view the real contents of the stored encrypted data. Further, the anonymity characteristic of blockchain prevents linking the preservation data to any individual user because all transactions use an Ethereum address which does not have the user’s personal information.

Table 3. Test Cases for Storage Efficiency of Health Records

- Parameter I: Storage efficiency of Health Records

Sr.No.	Test Case	Details	Result
1	Verify if the doctor can upload the medical record	Doctors can upload encrypted records after authentication	Successful
2	Ensure that the record cannot be downloaded.	The doctor cannot download but can analyze the shared record.	Successful
3	Patient should be able to see the records.	Patients can see their medical history from the records	Successful
4	Verify if the encrypted record can be effectively identified.	Encryption of the medical record with the session key and decryption with a private key is possible.	Successful

Table 4. Test cases for Level of Security

• Parameter II: Level of Security

Sr.No.	Test Case	Details	Result
1	Verify if the medical record is encrypted on IPFS	The medical record is encrypted using patient public keys and uploaded on IPFS and returns a hash value of the encrypted record	Successful
2	Verify if the public key infrastructure is used	Public and private keys are generated through a secured NTRU algorithm and used for encryption	Successful
3	Verify if the user maintains the session	The application session will be disconnected only when the user signs out or the session expires.	Successful
4	Verify that the encrypted record can be effectively identified	Encryption of the medical record with the session key and decryption with a private key is possible.	Successful

Table 5. Test cases for Improved data privacy

• Parameter III: Improved data privacy

Sr.No.	Test Case	Details	Result
1	Verify if the user can view the homepage and login based on the permission	When the url is exposed to the user, the homepage can see the login permission	Successful
2	Verify if the patient can share the required files with the valid doctors	Patient has granted permission to a specific doctor to maintain privacy	Successful
3	Verify if shared files are not downloadable	Shared file has only view access and updates the data with additional file	Successful

7.6. Test Cases

The performance of the proposed system was verified in terms of storage efficiency, security and privacy for various cases. The results are shown in Tables 3,4, and 5.

8. Conclusion

Privacy protection of personal data has been mandated as a fundamental right in many countries. The patient records are highly personal and require to be handled with due care to ensure the privacy of the individual giving full control to the patient and, at the same time, protecting from attacks by hackers. This paper proposes a novel system that applies a cryptographic process with self-sovereign architecture, employing blockchain technology to enhance data security and privacy. Using the IPFS database to store hash values makes the system tamper-resistant, and encrypted PKI provides efficient and secure access control. The NTRU algorithm is preferred due to its quantum-resistant characteristics and better throughput compared to other popular algorithms. Screenshots of the implementation have been provided, and the evaluated performance parameters of the proposed system are found to be encouraging. Further research will pave the way for real-time implementation of the proposed system with reduced cost.

Acknowledgments

The guidance, inspiration and encouragement received from Dr. Deven Shah and Mr. Ramani Iyer at various stages of implementation are gratefully acknowledged. This work would not have been possible without the support provided by Shah & Anchor Kutchhi Engineering College.

References

[1] Christian Esposito et al., “Blockchain: A Panacea for Healthcare Cloud-Based Data Security and Privacy?,” *IEEE Cloud Computing*, vol. 5, no. 1, pp. 31–37, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[2] Leadership for It Security & Privacy Across HSS Cyber security Program, Electronic Medical Records in Healthcare, 2022. [Online]. Available: https://www.hhs.gov/sites/default/files/2022-02-17-1300-emr-in-healthcare-tlpwhite.pdf

[3] Anna Kragie, ITRC Data Breach Report: Breaches up 17%, Exposed PII Records Down 41%, Rippleshot, 2020. [Online]. Available: https://info.rippleshot.com/blog/itrc-data-breach-report-pii-records#:~:text=Press-,ITRC%20Data%20Breach%20Report%3A%20Breaches%20up%2017%25%2C,Exposed%20PII%20Records%20Down%2041%25&text=The%20Identity%20Theft%20Resource%20Center's,%2C%20year%2Dafter%2Dyear.

[4] Ivan Homoliak et al., "The Security Reference Architecture for Blockchains: Towards a Standardized Model for Studying Vulnerabilities, Threats, and Defenses," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 341-390, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[5] Wie Liang Sim, Hui Na Chua, and Mohammad Tahir, "Blockchain for Identity Management: The Implications to Personal Data Protection," 2019 *IEEE Conference on Application, Information and Network Security (AINS)*, pp. 30–35, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[6] Smita Bansod, and Lata Ragha, "Blockchain Technology: Applications and Research Challenges," *International Conference for Emerging Technology (INCET)*, pp. 1–6, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[7] Nguyen Binh Truong et al., "GDPR-Compliant Personal Data Management: A Blockchain-Based Solution," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1746–1761, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[8] Jorge Bernal Bernabe et al., "Privacy-Preserving Solutions for Blockchain: Review and Challenges," *IEEE Access*, vol. 7, pp. 164908–164940, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[9] Wenxiu Ding, Zheng Yan, and Robert H. Deng, "Privacy-Preserving Data Processing with Flexible Access Control," *IEEE Transactions Dependable and Secure Computing*, vol. 17, no. 2, pp. 363–376, 2020, [CrossRef] [Google Scholar] [Publisher Link]

[10] Smita Bansod, and Lata Ragha "Challenges in Making Blockchain Privacy Compliant for the Digital World: Some Measures," *Sādhanā,* vol. 47, no. 168, 2022, doi: 10.1007/s12046-022-01931-1. [CrossRef] [Google Scholar] [Publisher Link]

[11] A. Z. Junejo, M. Ahmed, and A. Abdulrehman, "A Survey on Privacy Vulnerabilities in Permissionless Blockchains," *Article Published in International Journal of Advanced Computer Science and Applications(IJACSA)*, vol. 11, no. 9, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[12] John Preub Mattsson, Ben Smeets, and Erik Thormarker "Quantum-Resistant Cryptography," Cryptography and Security, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[13] Tiago M. Fernández-Caramès, and Paula Fraga-Lamas, "Towards Post-Quantum Blockchain: A Review on Blockchain Cryptography Resistant to Quantum Computing Attacks," *IEEE Access*, vol. 8, pp. 21091–21116, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] Keke Gai, Meikang Qiu, and Hui Zhao, "Privacy-Preserving Data Encryption Strategy for Big Data in Mobile Cloud Computing," *IEEE Transactions on Big Data*, vol. 7, no. 4, pp. 678-688, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[15] Shuyun Shi et al., "Applications of Blockchain in Ensuring the Security and Privacy of Electronic Health Record Systems: A Survey," *Computers & Security*, vol. 97, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[16] Alaa Haddad et al., "Blockchain for Healthcare Medical Records Management System with Sharing Control," *IEEE 7th International Conference on Smart Instrumentation, Measurement and Applications*, pp. 30–34, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[17] IPFS Community, IPFS doc - Privacy and encryption. [Online]. Available: https://docs.ipfs.io/concepts/privacy-and-encryption/

[18] Azeez Ajani Waheed et al., "An Integrated and Secured Web Based Electronic Health Record," *International Journal of Recent Engineering Science*, vol. 8, no. 4, 1 pp. 19-26, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[19] Ammar Ayman Battah et al., "Blockchain-Based Multi-Party Authorization for Accessing IPFS Encrypted Data," *IEEE Access*, vol. 8, pp. 196813–196825, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[20] Beena G Pillai, and Dayanand N Lal, "Blockchain-Based Asymmetric Searchable Encryption: A Comprehensive Survey," *International Journal of Engineering Trends and Technology*, vol. 70, no. 7, pp. 355–365, 2022. [CrossRef] [Publisher Link]

[21] Prasanna Ravi et al., "Lattice-based Key-sharing Schemes: A Survey," *ACM Computing Surveys*, vol. 54, no. 1, pp. 1–39, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[22] Praveen Gauravaram Tata, Harika Narumanchi, and Nitesh Emmadi, "Analytical Study of Implementation Issues of NTRU," *International Conference on Advances in Computing, Communications and Informatics*, pp. 700–707, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[23] Yakubov Alexander et al., "A blockchain-Based PKI Management Framework," *IEEE/IFIP Network Operations and Management Symposium*, pp. 1–6, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[24] Elie Kfoury, and David Khoury, "Distributed Public Key Infrastructure and PSK Exchange Based on Blockchain Technology," *IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData*), pp. 1116–1120, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[25] Jongbeen Han et al., "A Decentralized Document Management System Using Blockchain and Secret Sharing," *36th Annual ACM Symposium on Applied Computing*, pp. 305–308. 2021. [CrossRef] [Google Scholar] [Publisher Link]

[26] Guishan Dong et al., “Anonymous Cross-Domain Authentication Scheme for Medical PKI System,” *ACM Turing Celebration Conference*, no. 68, pp. 1–7, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[27] Nidhi, Dr. Arpinder Singh, "Steganography of ECG Signals for Hiding of Patient Confidential Data", *International Journal of Computer & organization Trends (IJCOT)*, vol. 5, no. 6, pp. 5-8, 2015. [CrossRef] [Publisher Link]

[28] Smita Bansod, and Lata L. Ragha, “*Blockchain Impact of Security and Privacy in Digital Identity Management*,” Blockchain for Information Security and Privacy, Auerbach Publications, pp. 275–291. 2021. [Google Scholar] [Publisher Link]

[29] S. Bansod, and L. Ragha, “Secured and Quantum Resistant Key Exchange Cryptography Methods – A Comparison,” *Interdisciplinary Research in Technology and Management (IRTM)*, pp. 1–5. 2022. [CrossRef] [Google Scholar] [Publisher Link]

[30] National Institute of Standards and Technology , FIPS 197, Advanced Encryption Standard (AES), NIST, 2001. [Online]. Available : https://csrc.nist.gov/publications/detail/fips/197/final

[31] Shuai Wang et al., “Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems.*, vol. 49, no. 11, pp. 2266–2277, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[32] Philip S. Hirschhorn et al., “Choosing NTRUEncrypt Parameters in Light of Combined Lattice Reduction and MITM Approaches,” *Applied Cryptography and Network Security*, vol. 5536. pp. 437–455, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[33] V. Mavroeidis et al., “The Impact of Quantum Computing on Present Cryptography*,” International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 9, no. 3, pp. 405-414, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[34] Yufei Lin, and Chongyang Zhang, “A Method for Protecting Private Data in IPFS,” *IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 404–409, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[35] Christos Patsonakis et al., “Implementing a Smart Contract PKI,” *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1425–1443, 2020. [CrossRef] [Google Scholar] [Publisher Link]