FortiRx: Distributed Ledger based Verifiable and Trustworthy Electronic Prescription Sharing

Anand Kumar Bapatla[1][0000−0003−4567−7827],
Saraju P. Mohanty[1][0000−0003−2959−6541], and Elias Kougianos[2][0000−0002−1616−7628]

[1] Department of Computer Science and Engineering, University of North Texas, USA.
anandkumarbapatla@my.unt.edu, saraju.mohanty@unt.edu
[2] Department of Electrical Engineering, University of North Texas, USA.
elias.kougianos@unt.edu

Abstract. A paper-based prescription signed by the prescriber to authorize dispensing of medication is typically used in traditional healthcare. Such systems are prone to many issues like medication errors, latency, and lack of integration with other healthcare systems. Hence, Electronic prescription (E-prescription) systems are being used as alternatives to overcome these issues. Even though E-prescription systems provide the advantage of recording and maintaining patient medication history but still face issues such as system crashes, latency due to their centralized architectures, prone to many security threats like identity theft and unauthorized patient record access and modifications. Lack of standardization can also make such E-prescription systems not interoperable, which may lead to information fragmentation or delays in the processing of prescriptions. Hence, there is still a need for making these E-prescription systems more secure, reliable, and cost-effective for wide-range adaptation. Blockchain is one such technology that can add additional layers of security to the existing E-prescription systems by providing tamper-proof records of all transactions which will help in ensuring the authenticity and integrity of prescriptions. Blockchain can also help in better management of patients' privacy while patients still have full control over their health data. Blockchain usage can also enhance interoperability and reduce prescription abuse. The proposed application FortiRx makes use of the Ethereum blockchain platform and leverages smart contracts for implementing business logic. Cyphertext-Policy Attribute-Based Encryption (CP-ABE) is used in the proposed application to create and manage access-control mechanisms and ensure Health Insurance Portability and Accountability Act (HIPPA) compliance. The proposed system has been implemented and analyzed for security, reliability, and adaptability in a real-time environment.

Keywords: Smart Healthcare · Healthcare Cyber-Physical System (H-CPS) · Electronic Prescription · Blockchain · Distributed Ledger · Ethereum · Smart Contracts · Attribute Based Encryption · Cyphertext-Policy Attribute Based Encryption

1 Introduction

Paper-based prescriptions have been one of the preferred ways of sending medication instructions by physicians for a long time. Handwritten prescriptions are often difficult to read and susceptible to fraud by duplication. It is also a time-consuming process that requires patients to physically deliver the prescription to pharmacies. It also causes issues while integrating with

Electronic Health Records (EHR) and makes it difficult to keep track of patient's medication history. All these issues with traditional paper-based prescriptions cause medication errors [1]. Medication errors are defined as preventable events which cause improper usage of medication by patients and can lead to patient harm. These medication errors can occur at multiple aspects of medication systems such as while prescribing the drugs, while entering prescription information into the electronic systems, while preparing the prescription for dispense, or while consuming the medication by patients [8]. The World Health Organization (WHO) determined medication errors are the leading cause of patient harm and account for up to $42 billion annually [16, 21, 24]. In order to reduce the number of medication errors, electronic prescription systems have been put into place to make the whole process computerized to reduce human errors. Initially, the patient requests a refill, or the physician creates a new prescription record based on the diagnosis of the patient and enters it into the electronic record system. Electronic Medical Record (EMR) or Electronic Health Record (EHR) system then generates an electronic prescription transaction and with the prescription along with the physician information to a centralized E-prescription system. E-prescription then verifies the authenticity of the prescription before sending it to the authorized pharmacies from which the patient will be able to pick it up. Any refill requests will be generated by the pharmacy and sent back to the physician using the E-prescribing system and notify the patient for pickup. The working of the E-prescribing system can be clearly seen in Figure 1. Other entities like government agencies U.S. Food and Drug Administration (FDA), Centers for Medicare and Medicaid Services (CMS), Healthcare Information Exchanges (HIEs) along with Pharmacy Benefit Managers (PBMs) are also involved in E-prescribing systems. Due to the complex nature of interactions, those are not shown in Fig. 1.

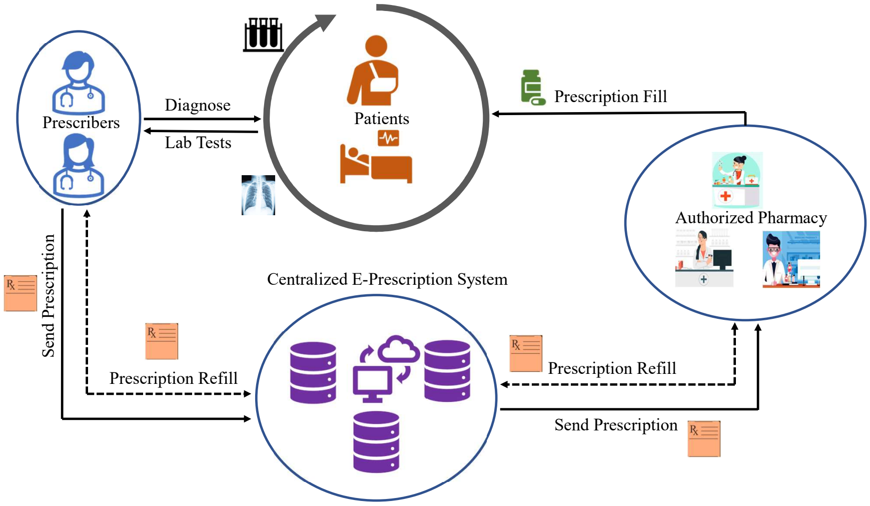


Fig. 1. E-Prescribing System Working

E-prescription is the ability to use the computerized system by healthcare providers to create, manage and transmit prescription data to pharmacies. The E-prescription system increases the legibility of prescriptions thereby reducing the likelihood of errors. It also increases efficiency by automating most of the processes and reducing the administrative processing needed to issue and manage paper-based prescriptions. By automating the processes and reducing medical errors, costs associated with medical errors can also be avoided. It also makes it easy to integrate with EMR systems which can help in managing patient health records more efficiently. Although E-prescription systems provide a variety of benefits compared to the paper-based approach, it still faces many issues which need to be addressed.

The centralized architecture of the E-prescription can lead to latency in processing large amounts of information and even sometimes lead to a Single Point of Failure (SPOF). They are even prone to many other security threats that may lead to data privacy and security issues [26]. This lack of standardization between such centralized systems can lead to problems during integration with Electronic Healthcare Record Systems (EHRs). The cost of deployment and maintenance of such systems is also a major hurdle for adopting. The complexity of such systems is another problem for users of e-prescription systems.

The rest of the paper is organized as follows: Section 2 presents possibility of using blockchain or distributed ledger for e-prescription in H-CPS. Section 3 gives an overview of prior related research. Section 4 talks about the novel features of the proposed FortiRx system. Section 5 discussed the preliminaries and Section 6 describes the architecture of the proposed FortiRx system. Section 7 discusses proposed algorithms for uploading and managing prescription information. Section 8 describes the implementation of the proposed FortiRx system. Section 9 provides the analysis details and Section 10 provides conclusions along with future research aspects.

2 Blockchain As A Solution for Robust E-Prescription

Blockchain is a type of distributed ledger technology that helps in recording and storing information in a secure, transparent, and distributed manner in Peer-to-Peer (P2P) network. Blockchain is initially designed for digital assets in Bitcoin [15] but this technology has revolutionized and shown potential use-cases in many domains including patient EHR Management [9, 10], supply chain management in the pharmaceutical industry and medicine manufacturing [4, 13] etc. The main components of the blockchain consist of Distributed Ledger (DL), Nodes, Transactions, and consensus protocol. Block ledger generally consists of hash-connected blocks which have both header with all metadata and body with actual transactions. Header and body are implemented so that the light nodes which doesn't want to store all the transaction data can prune the body and just store the header information and still verify any transaction. The structure of the ledger varies for different implementations and the most prominent structures are blockchain which is a linear structure of blocks and Decentralized Acyclic Graph (DAG) which uses graph structure for scalability of the network. All the participants in the P2P network are called nodes and are classified as light node, miner node, and full nodes based on their roles and responsibilities. Light nodes are the nodes with limited storage and computational capabilities which don't store the entire history of the transaction data, usually, these are the participant nodes that try to utilize blockchain infrastructure to perform transactions. Mining nodes are special nodes that have the large computational power and usually compete to solve a hard mathematical problem to win a chance of adding a new block to the ledger. Miners are awarded block rewards combined with transaction fees from the transactions included in the block generated by them. Full nodes are responsible for storing complete copies of the ledger including all transaction data and verifying both transactions and blocks which ensures the security and integrity of the blockchain. Usually, full nodes are not compensated with any fees or rewards. Communication in a P2P network without delegation of centralized authority is prone to issues of disagreement which is described as a Byzantine General Problem [12]. Consensus protocols are used in blockchain to reach a consistent state of the system where all or majority of the nodes accept the validity of the block even with some of the nodes in networks acting maliciously. Some of the most prominent consensus protocols include Proof-of-Work (PoW), Proof-of-Stake (PoS), Proof-of-Authority (PoA), etc. Some of the features provided by blockchain include Decentralization, Security, Transparency, Immutability, and Faster processing by removing centralized entities.

Evaluating E-Prescription System Against Blockchain Even though blockchain provides many features, it cannot be a solution for every application. Hence each application has to satisfy certain conditions for applying blockchain [17].

Criteria: Does the current application need tamper-proof permanent storage?
Evaluation: E-prescription systems are intended to store the prescription information of the patients for enabling them to keep a record of their medication and to avoid adverse interactions of medications prescribed for other conditions. As this application needs permanent storage, blockchain can be a solution.

Criteria: Are there multiple trust-less data contributors to the data?
Evaluation: E-prescription systems consist of many distributed entities which include a network of prescribers, a network of patients, a network of pharmacies, and other regulating bodies. All these entities are distributed geographically and don't trust each other and need a co-coordinating central authority in order to relay. information. As there are multiple trustless data contributors in the network, blockchain can be a solution.

Criteria: Does the application modify data after storage?
Evaluation: Issuing prescriptions, and dispensing prescribed medicines generally don't need modifications once processed and prescribed medicines are dispensed to patients. If there are any mistakes in the prescription, a new prescription should be sent while invalidating the previous prescription. Hence, blockchain can be an acceptable solution in this case.

Criteria: Is data privacy required?
Evaluation: According to the Health Insurance Portability and Accountability Act of 1996 (HIPAA) [7], sensitive patient information shouldn't be disclosed without the patient's consent. Blockchain as such doesn't provide privacy on shared data as the data will be stored at multiple nodes in the network. But in the current application, we have used Cipher Text Policy - Attribute Based Encryption (CP-ABE) which will enable patients with full control over their prescription information and only encrypted data is being shared using off-chain storage Inter Planetary File System (IPFS).

Features of blockchain technology can provide an additional layer of security for the E-Prescription system by creating an immutable log of all transaction which help in maintaining the integrity of prescription data. Immutable log created acts as a single source of truth which is available at all the distributed participating entities and help to reduce prescription fraud. Blockchains also enhance interoperability by enabling different healthcare providers to participate and share data in the network transparently. As digital wallets are used for performing transactions on a blockchain network, the privacy of patients is maintained which will prevent data leakages.

3 Related Prior Works

Blockchain is one of the recent technological advancements which was introduced primarily for digital currency systems which enable P2P money transfers that are faster, and more reliable, and at any point in time digital assets will be under the control of the owner instead of centralized entities like banks. The features of blockchain including transparency, data integrity and security, working in a trust-less environment, and consensus-based updates have shown promising solutions in many other fields. Whether it is to store and manage Electronic Health Records [18, 22] or secure medical supply chain [2, 4–6] healthcare industry has been benefited largely by adapting blockchain technology. Different studies are conducted to build E-Prescription by leveraging blockchains. Comparison of proposed FortiRx with state-of-art is clearly shown in Table 1.

Table 1. Comparative Analysis of Proposed FortiRx with state-of-art.

	Blockchain Platform	**Smart Contracts**	**Off-chain storage**	**Data Privacy**	**Access Control Mechanism**	**CP-ABE**
Thatcher, et al. 2018 [20]	Ethereum	✓	✗	✗	✗	✗
Musamih, et al. 2021 [14]	Ethereum	✓	✓	✗	✓	✗
Taylor, et al. 2022 [19]	Ethereum	✓	✗	✓	✓	✗
Alnuaimi, et al. 2022 [3]	Ethereum	✓	✓	✗	✓	✗
Ionescu, et al. 2022 [11]	Ethereum	✓	✗	✗	✗	✗
FortiRx (Current Paper)	Ethereum	✓	✓	✓	✓	✓

A use case of blockchain as a Prescription Drug Monitoring System (PDMP) is presented in [20]. This proposed system leverages smart contracts on the Ethereum platform for the creation and management of prescriptions. Even though the proposed mechanism is providing a solution to manage E-Prescriptions, it mainly uses on-chain storage which could be very expensive for large amounts of information, and no access control mechanism is defined which will create a lot of security threats in the system. Another application of blockchains in prescription drug supply is given in [25]. This work provides a dynamic identity mechanism to prevent patient privacy issues along with a robust authentication protocol combined with blockchain. However, the proposed architecture doesn't deal with prescription management. Another blockchain-based solution for controlled medication is discussed in [14]. In this work, smart contracts and the Ethereum platform are used for building the controlled medicine prescription system along with using IPFS as off-chain storage. However, No access control mechanism is implemented in IPFS which will make data available for all the participants in the IPFS network.

VigilRx proposed in [19] makes use of smart contracts and the Ethereum platform to create a system for creating and managing prescriptions, it also implemented a robust access control mechanism using the RBAC mechanism. However, prescription information is stored on-chain which can be difficult to manage and is not cost-effective. Similar to previously proposed solutions, blockchain-based health insurance claims for prescription drugs are proposed in [3] which leverages the Ethereum blockchain and smart contracts. In this proposed architecture IPFS is used as a solution for storing prescription data to avoid on-chain overhead. Even though an efficient mechanism for managing insurance claims is proposed, Prescription information stored in IPFS is prone to data privacy risk as the information will be stored at all the participating public nodes in the IPFS network. Another implementation of the Ethereum platform and smart contracts can be seen in [11]. However, this proposed system doesn't address both data privacy concerns and the overhead costs of on-chain data storage.

4 Novel Contributions of the Current Paper

Below are problems with centralized E-Prescription systems which are addressed in the proposed FortiRx architecture along with novel solutions proposed.

4.1 Problems with Centralized E-Prescription System Addressed in FortiRx

Problems with existing centralized e-prescription systems which are addressed by novel proposed FortiRx architecture are:

- Centralized architecture in E-prescription systems can lead to a Single Point of Failure (SPOF).
- As the number of transactions increases, latency increases significantly.
- Lack of standardization can cause difficulty in integrating the systems with other Electronic Health Record Systems (EHR).
- Centralized systems are more prone to security threats and lead to other forms of fraud and prescription abuse.
- Cost of deployment and maintenance of E-prescribing systems is huge and makes it not accessible for all healthcare providers.
- Complexity of using such technology can also resist some people from adopting e-prescription systems.

4.2 Novel Solutions Proposed

The novel contributions of the proposed FortiRx are:

- Proposed FortiRx makes use of blockchain combined with the distributed file system (IPFS) to create a decentralized environment for all the participating entities to share prescription data.

- Blockchain creates a distributed trust-less environment to share data which enhances the interoperability of the system.
- Usage of distributed file-sharing system to store prescription information can help in reducing the amount of on-chain data.
- Due to the decentralized nature of the proposed FortiRx, It is resistant to Single Point of Failure (SPOF) and also reduces response latency.
- It avoids data tampering and prescription abuse by maintaining a distributed ledger that acts as a single source of truth.
- Proposed FortiRx also makes use of Cipher text-Policy attribute-based encryption (CP-ABE) to provide a robust access control mechanism.

5 Preliminaries of Cipher text-Policy Attribute-based Encryption (CP-ABE)

5.1 Bilinear Map

A bilinear map in CP-ABE are called pairings which associates pairs of elements from two groups G_1 and G_2 to a third group G_T. When G_1=G_2 the pairing is called symmetric otherwise asymmetric. Such a map function is denoted as $e:G_1\times G_2\rightarrow G_T$ and must satisfy the following properties:

- Bilinearity: For all $a,a'\in G_1$ and $b,b'\in G_2$, we have:

$$e(a+a',b)=e(a,b)\cdot e(a',b) \tag{1}$$

and

$$e(a,b+b')=e(a,b)\cdot e(a,b') \tag{2}$$

- Non-degeneracy: There exist generators g_1 and g_2 of G_1 and G_2, respectively, such that $e(g_1,g_2)$ is a generator of G_T.
- Computability: The bilinear map function e can be efficiently computed.

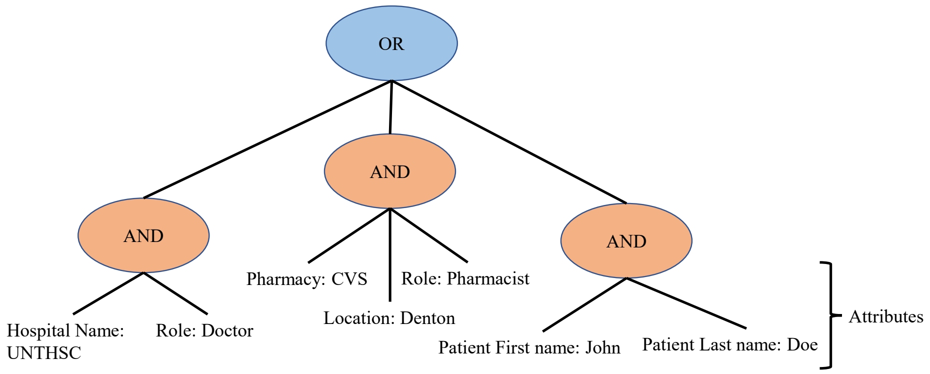
Bilinear map is used in CP-ABE to compute pairing of two elements $g_1^a\in G_1$ and $g_2^b\in G_2$. This is done by using the equation:

$$e(g_1^a,g_2^b)=e(g_1,g_2)^{ab} \tag{3}$$

5.2 Setup

During setup, a trusted entity generates the system parameters which include the master secret key, a description of the bilinear map, and the groups used in the encryption and decryption process. A master secret key generated is used to generate the private keys for different attributes. Below are the steps followed in the setup function:

- Choose a Elliptic curve E and two cyclic groups G_1 and G_2 of order q such that $E(G_1,E_2)\subseteq G_T$.
- Choose corresponding generators g_1 and g_2 for two groups G_1 and G_2 respectively.
- Choose a master secret key randomly $a\in\mathbb{Z}_q$.
- Generate the set A = a * g2.
- publish the generated system parameters (E,G_1,G_2,G_T,e(g_1,g_2),A) and master secret key a is kept secret at the trusted entity.



Access Policy = ((“Hospital Name”== “UNTHSC” AND “Role”== “Doctor”) OR
(“Pharmacy”==“CVS” AND “Location”==“Denton” AND “Role”==“Pharmacist”) OR
(“Patient First Name”==“John” AND “Patient Last Name”==“Doe”))

Fig. 2. Sample Access Policy for Prescription.

5.3 Key Generation

A set of attributes are assigned to a user which is used to generate a private key for that user. The set of attributes associated with the user will be sent to the trusted entity which makes use of the master secret key along with the set of attributes to generate a decryption key.

5.4 Encryption

To encrypt a message using CP-ABE, the plain text message is first mapped to the elements in the encryption group. Next, a policy is defined using a set of attributes required to decrypt the cipher text. The defined policy is then mapped elements in the encryption group. After that plain text is then encrypted using the policy as an additional parameter to generate cipher text.

5.5 Decryption

A user with an appropriate set of attributes will first retrieve the decryption key and then uses the decryption key to decrypt the cipher text to plain text.

5.6 Access Policy

The access policy in CP-ABE is a logical expression defined using Boolean operators AND, OR, and NOT that specifies the attributes which are required to decrypt the ciphertext. Each set of attributes is associated with a secret key and all the cipher texts with access policy satisfying these attributes will be able to decrypt. A sample access policy is shown in Fig 2.

6 Architectural Overview of the Proposed FortiRx

System level architecture view of the proposed FortiRx is shown in Fig 3. Many distributed entities participate and share data in the proposed FortiRx which include prescribers, patients,

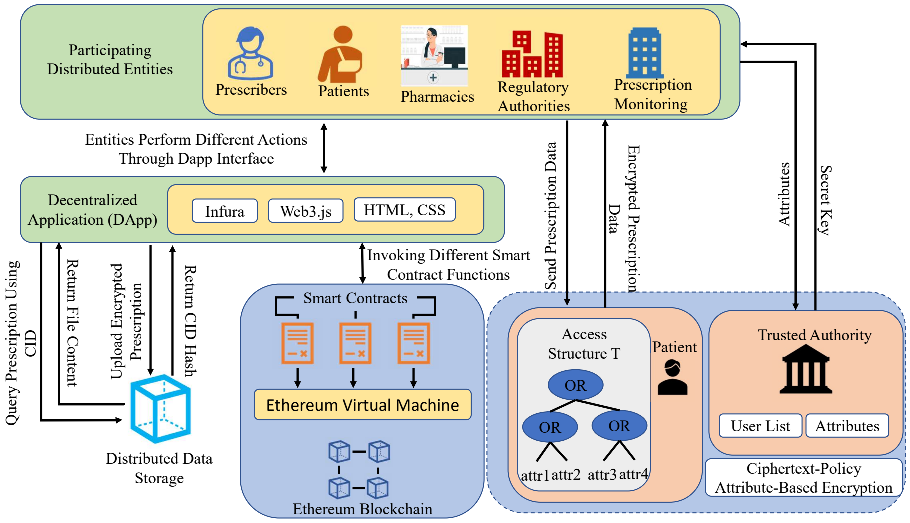


Fig. 3. Architectural Overview of Proposed FortiRx System.

pharmacies, Regulatory authorities, and in some cases authorities from prescription monitoring for controlled substance prescriptions. Prescriber is responsible for creating digital prescriptions information and encrypting them before sending the files to distributed data storage. Access Policy for encryption is provided by the patients that will determine who is allowed to access the data based on the issue attributed by a trusted authority.

One of the major problems with blockchain is managing large amounts of data on-chain which can be expensive too. Prescription systems generate large amounts of data within a short span of time and managing such data on-chain is not a viable solution. Hence, in the proposed FortiRx, we have used an off-chain distributed data storage solution to handle all the prescription information. Even though data is stored at all the decentralized nodes, implemented CP-ABE mechanism will make sure only the entities having properly assigned attributes defined in the access policy can decrypt it which ensures data privacy.

The proposed FortiRx architecture leverages smart contracts to implement business logic which include issuing the prescriptions, updating the status of prescription, requesting refills, approval for refills, and verifying the authenticity of prescription. Apart from business logic, a robust Role-based Access Control mechanism (RBAC) is also designed to ensure the entities can access only allowed functions. Every transaction by any of the entities participating will generate an event in the blockchain and will be added to the immutable log. Participating entities will access the functions of smart contracts through Decentralized Application (DApp) designed.

7 Proposed Algorithms for FortiRx

The prescription creation and uploading process is shown in Algorithm 1. After performing the diagnosis, an electronic prescription file will be generated with all the prescription information along with the patient information and dosages. This file will be read and the file content

Algorithm 1 Proposed Prescription Upload Algorithm for FortiRx.

```
Input: Digital Prescription Data, public parameters $(params, g_1, g_2, e)$ generated during CP-ABE
    setup, Access policy $\rho$ defined by the patient
Output: Content ID for IPFS file, Transaction hash of prescription creation in blockchain
A digital prescription is generated, and a file is created
for Each prescription file f do
    Open file in read mode
    FileItem $\leftarrow$ open(filePath,'r')
    Read prescription content from the file
    prescription content $(P_{content})$ $\leftarrow$ fileItem.read()
    Encryption is done using public key (pk) and policy $\rho$ to generate ciphertext of the prescription
    content
    Cipher text $CT$ $\leftarrow$ cpabe.encrypt(pk, $P_{content}$ ,$\rho$)
    New file is created and generated cipher text is written to that file
end for
for Each encrypted prescription file f do
    Send upload request to IPFS
    Response (res) $\leftarrow$ requests.post(Infura end point, authentication parameters, file f)
    Content ID from response is retrieved
    Content ID (CID) $\leftarrow$ res.text['Hash']
end for
Prescriber creates a new createPrescription transaction in prescription smart contract
Transaction $(Tx)$ $\leftarrow$ prescription.createPrescription(patient address $(P_{addr})$,CID)
if caller == Prescriber then
    New prescription is created and added to patient address
    Emit an event (ev) with prescription data and a log is generated
    Return transaction hash $(Tx_{hash})$
else
    Reject $Tx$
end if
```

is encrypted by using the public parameters generated during the setup process of CP-ABE along with the access policy which will be decided by the patient. Once the encryption process is done, the generated cipher text will be written to a file and uploaded to the distributed data storage (IPFS). A content ID will be returned once the file upload is successful which is useful during the retrieval process. Prescriber usually the physician then creates a transaction using the patient's Ethereum address and content ID from IPFS to create a prescription entry in the blockchain. RBAC mechanism implemented in the smart contract will ensure the transaction is coming from the actual prescriber before updating the details of the patient account. Once the transaction is successful, the transaction hash and generated prescription ID will be sent back to the called and in case of any errors, the transaction will be discarded.

Retrieval of prescription data from IPFS and decryption is clearly explained in Algorithm 2. The Entity trying to access will make a function call to the smart contract using the prescription ID generated during the upload process. Based on the prescription ID, details of the prescription along with the IPFS hash are retrieved. This IPFS hash which is also the content ID will be used to request IPFS for retrieving the prescription content. Retrieved prescription content is an encrypted string and to decrypt it, the entity needs the secret key. The entity then sends its attributes to the trusted authority to get the secret key to decrypt the prescription content.

Algorithm 2 Proposed Prescription Retrieval Algorithm for FortiRx.

```
Input: Prescription ID ($P_{ID}$) generated while creating new prescription in blockchain, attribute list
    of requesting entity ($attr\_list$)
Output: Decrypted prescription content ($P_{content}$)
for Each view request ($req$) do
    Send a function call to prescription smart contract to retrieve Prescription based on $P_{ID}$
    Retrieved prescription $P_{ret} \leftarrow$ prescription.viewPrescription($P_{ID}$)
    Get IPFS Hash (CID) from the function response
    CID $\leftarrow P_{ret}$['IPFSHash']
    Send request to IPFS to retrieve prescription content ($P_{content}$)
    Response (res) $\leftarrow$ requests.post(Infura end point, CID, authentication parameters)
    Retrieved cipher text ($CT$) $\leftarrow$ res.text
    Secret key for set of attributes $attr\_list$ is requested from trusted authority
    Secret key ($Sk$) $\leftarrow$ cpabe.keygen(public key (pk), $attr\_list$)
    Decrypt cipher text using the secret key to get prescription content
    if $\rho$.evaluate($attr\_list$) then
        $P_{content} \leftarrow$ cpabe.decrypt($Sk$, $CT$)
    else
        Cannot decrypt prescription content
    end if
end for
```

If attributes satisfy the policy defined during the encryption, the content will be decoded and accessible to the requesting entity. If not, the content cannot be decrypted ensuring data privacy.

The status of the prescription is also updated once medicines are dispensed by the pharmacy. The pharmacy creates a transaction call to the prescription smart contract with the prescription id as the parameter. RBAC mechanism implemented will ensure the requesting Ethereum address is assigned a pharmacy role and it is one of the registered pharmacies for the patient. Similarly, in the case of refills, the pharmacy will update the status of the prescription requestRefill flag and notify the prescribers to approve. Different steps involved in this process are shown in Algorithm 3

8 Implementation of FortiRx

8.1 Smart Contract Design

We have used solidity language to design smart contracts and deployed them in the Ethereum platform. As there are multiple entities with each one having its own functions to perform, the access control mechanism is important. We have developed a Role Based Access Control Mechanism (RBAC) using smart contract functions and modifiers. For each type of entity: Patient, Prescriber, and Pharmacy, three role smart contracts are defined which have different functions to add users to the role, check if the given address has been assigned with a role, and revoke the roles. Along with these modifiers are also defined to check for attaching them to the functions. modifiers are used to impose some pre and post-conditional checks on the function parameters passed.

The main business functionality of prescription creation and management is done in ”FortiRx.sol” smart contract which has two mappers which are used to keep track of prescriptions

Algorithm 3 Status Updates for Prescription on Blockchain.

```
Input: Prescription ID ($P_{ID}$) generated while creating new prescription in blockchain
Output: The Status of the prescription will be updated
Different status flag updates will be sent either by pharmacy or physician
Based on the type of status update, different functions of the smart contract will be invoked with ($P_{ID}$) as parameter
if Prescription is filled then
    prescription.updatePrescriptionStatus($P_{ID}$)
    Smart contract check the pharmacy Ethereum address for access and updates isFilled flag of prescription
else if Prescription needs re-filling then
    prescription.requestRefill($P_{ID}$)
    Smart contract checks the pharmacy Ethereum address for access and updates the requestRefill flag of prescription
else
    prescription.issueRefill($P_{ID}$)
    Smart contract checks the physician's Ethereum address for access and updates the isFilled and requestRefill flags of prescription
end if
```

generated and another one to associate prescriptions to the patient's Ethereum address which will make it easy to retrieve. A function is developed for creating new prescriptions, this is restricted only to prescribers who are usually physicians. This creates new prescription functions and takes the patient address and Content ID of the prescription uploaded to IPFS as parameters. Another important function is the update prescription status function which is restricted to role of a pharmacy. Once the prescription is dispensed, the pharmacy will call this function to update the prescription to be filled and creates a log in the blockchain. In case of refill is required, the pharmacy can also make a call to function request refill which takes the prescription id as a parameter and updates the flag for a refill. On checking the flag, prescribe can approve the refill. A view function is also created to retrieve and view prescription details (Not the content of the prescription and patient information). Complete Class diagram of designed smart contracts for proposed FortiRx can be seen in Fig 4.

8.2 Blockchain Network

The proposed FortiRx system is implemented using the Ethereum platform leveraging smart contracts designed in solidity language. A decentralized application (DApp) is designed using the truffle framework and a functionality test is performed using chai. Implemented FortiRx is first deployed in the local Ganache blockchain which mimics the working of Ethereum mainnet but provides 10 free accounts with test ether of 100 ETH each. For measuring performance analysis of the implemented system, it is then deployed in Ethereum testnet Sepolia. Sepolia is a permissioned Proof-of-Stake (PoS) consensus-based testnet. As it is a public testnet and hosts many other DApps with live transactions happening, evaluating FortiRx can give better results to evaluate reliability and adaptability in a real-world deployment scenario.

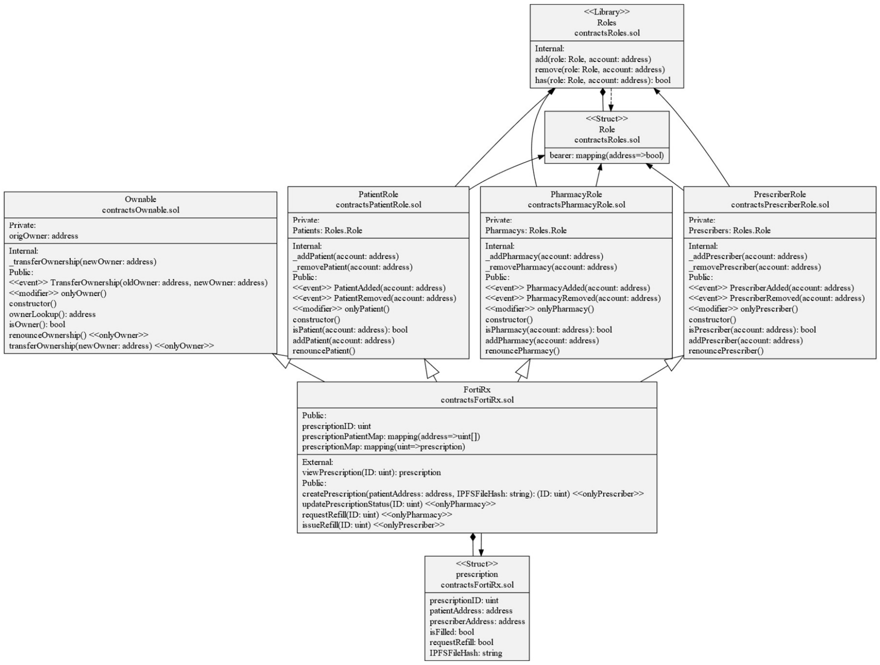

<<Library>>
Roles
contractsRoles.sol
Internal:
add(role: Role, account: address)
remove(role: Role, account: address)
has(role: Role, account: address): bool
<<Struct>>
Role
contractsRoles.sol
bearer: mapping(address=>bool)
Ownable
contractsOwnable.sol
Private:
origOwner: address
Internal:
_transferOwnership(newOwner: address)
Public:
<<event>> TransferOwnership(oldOwner: address, newOwner: address)
<<modifier>> onlyOwner()
constructor()
ownerLookup(): address
isOwner(): bool
renounceOwnership() <<onlyOwner>>
transferOwnership(newOwner: address) <<onlyOwner>>
PatientRole
contractsPatientRole.sol
Private:
Patients: Roles.Role
Internal:
_addPatient(account: address)
_removePatient(account: address)
Public:
<<event>> PatientAdded(account: address)
<<event>> PatientRemoved(account: address)
<<modifier>> onlyPatient()
constructor()
isPatient(account: address): bool
addPatient(account: address)
renouncePatient()
PharmacyRole
contractsPharmacyRole.sol
Private:
Pharmacys: Roles.Role
Internal:
_addPharmacy(account: address)
_removePharmacy(account: address)
Public:
<<event>> PharmacyAdded(account: address)
<<event>> PharmacyRemoved(account: address)
<<modifier>> onlyPharmacy()
constructor()
isPharmacy(account: address): bool
addPharmacy(account: address)
renouncePharmacy()
PrescriberRole
contractsPrescriberRole.sol
Private:
Prescribers: Roles.Role
Internal:
_addPrescriber(account: address)
_removePrescriber(account: address)
Public:
<<event>> PrescriberAdded(account: address)
<<event>> PrescriberRemoved(account: address)
<<modifier>> onlyPrescriber()
constructor()
isPrescriber(account: address): bool
addPrescriber(account: address)
renouncePrescriber()
FortiRx
contractsFortiRx.sol
Public:
prescriptionID: uint
prescriptionPatientMap: mapping(address=>uint[])
prescriptionMap: mapping(uint=>prescription)
External:
viewPrescription(ID: uint): prescription
Public:
createPrescription(patientAddress: address, IPFSFileHash: string): (ID: uint) <<onlyPrescriber>>
updatePrescriptionStatus(ID: uint) <<onlyPharmacy>>
requestRefill(ID: uint) <<onlyPharmacy>>
issueRefill(ID: uint) <<onlyPrescriber>>
<<Struct>>
prescription
contractsFortiRx.sol
prescriptionID: uint
patientAddress: address
prescriberAddress: address
isFilled: bool
requestRefill: bool
IPFSFileHash: string

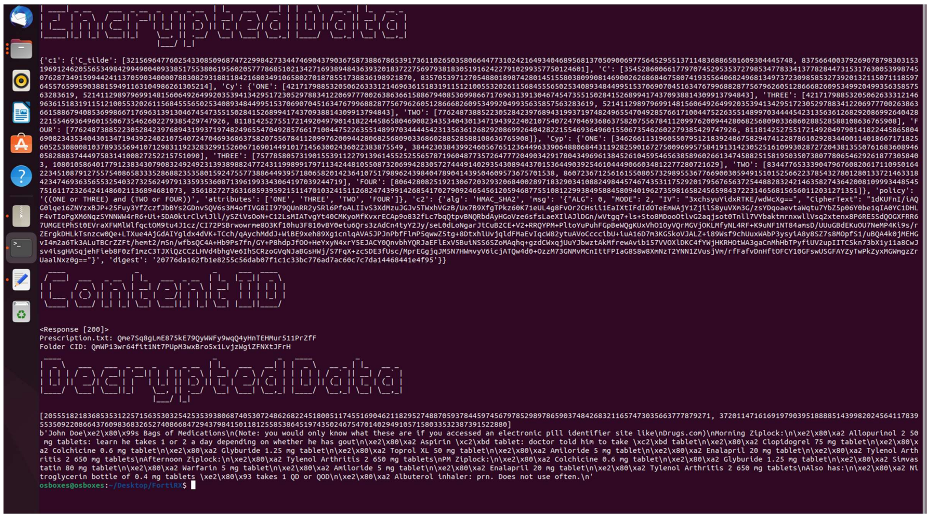
Fig. 4. UML Diagram of Implemented FortiRx Role and Prescription Smart Contracts

8.3 Distributed Data Source

Inter-Planetary File System (IPFS) is used for providing distributed data sources to store all the prescription information. As discussed before, storage is expensive and hard to manage on-chain. As the prescription application needs large amounts of information to be stored and retrieved, the off-chain distributed data storage solution IPFS is used. As it is impractical for all the users of blockchain to host their own nodes to participate in the network, we have used the infura platform which provides tools to connect to Ethereum easily and perform transactions. Infura also provides tools to connect to IPFS without hosting a node. IPFS connection uses the authentication parameters to authenticate the user before uploading the files.

8.4 CP-ABE System Design

For implementing the CP-ABE scheme, Ubuntu 22.04 64-bit with a base memory of 4GB of memory is hosted in a virtual environment. For prototyping the proposed CP-ABE scheme for FortiRx, the charm crypto library is used. Charm is a framework that is designed to rapidly prototype cryptosystems and implement different schemes. GNU Multiple Precision Arithmetic Library (GMP), Pairing-Based Cryptography Library (PBC), and OpenSSL are prerequisites for installing the charm framework. CP-ABE scheme proposed in [23] is used for implementing FortiRx.



```
EncryptedData

{'c1': {'C_tilde': [32156964776025433085096874722998427334474690437903675873886786539173611026503580664477310242164934046895681370509006977564529551371148368865016093044457 48, 837566400379269078798303153
1969124620556534984299490004093385175538061956020577786851021342716938948436393201837227569793818305191624227910299357750124601], 'C': [35452860066177970745295353727985347783341377828447315317630053998745
07628734915994424113705903400007883082931881184216803491065802701878551738836198921870, 8357053971270548801898742801451558038099081469002626868467580741935564068249681349737230985853273920132115071118597
645576599590388159491163104986261305214], 'Cy': {'ONE': [42171798853205062633312146963615183191151210055320261156845556502534089348449951537069070451634767996882877567962605128666826095349920499356358575
63283619, 521411298979699148156064926499203539413429517230529788341220697770026386366158867940853699866717696313913046745473551502841526899417437093881430991379484 3], 'THREE': [42171798853205062633312146
9636151831911512100553202611568455565025340893484499515370690704516347679968828775679626051286668260953499204993563585756328 3619, 5214112989796991481560649264992035394134295172305297883412206977700263863
661588679408536998667176963139130467454735515028415268994174370938814309913794843], 'TWO': [776248738852230528423976894319937197482496554704928576617100447522635514899703444454231356361268292086992640428
221554693649601550673546260227938542974792 6, 811814252755172149204979014182244586580469082343534043013471943922402107540724704693686375820755678411209976200944280682568090336860288528588108636765908], 'F
OUR': [7762487388522305284239768943199371974824965547049285766171004475226355148997034444542313563612682920869926404282215546936496015506735462602279385429747926, 8118142527551721492049790141822445865804
690823435340430134719439224021075407247046936863758207556784112099762009442806825680903368602885285881086367659 08]}, 'Cyp': {'ONE': [346266113196055079512183924867582947412287861029283440011401867171825
605253080081037893556941071298311923283299152606716901449101714563002436022383875549, 384423038439924605676512364496339064880684431192825901672750096995758419131423052516109930287272043813550761683608946
0582888374449758314100827252215751090], 'THREE': [75778580573190155391122791396145525255657871960487735726477720490342917804349696138452610459546563858960266134745882515819503507380778065462926187730584 0
3, 1080105864017791238343079083249249231393898824772431199899179711342448105508732069942830572744491402935430894437015364490392546104449066034812277280721629], 'TWO': [83447765333904796760820617110950164
22345108791275575408658333528688235358015924755773886449395718065820142364107517989624398404789041439504609573675701538, 860723671256161550805732989553677669003059491510152566223785432780128013372146331 8
4234746936356553254032732562497913359353608713961993343064197039244719], 'FOUR': [80642808251921306720329320684002897183290341088249844574674353117529201795676563725448828324214635827436420081099934485 45
751611723264241486021136894681073, 356182727363168593959215114701032415112682474399142685417027909246545612059468775510812299384958845809401962735981658245659843722314656815656011203127135 1]}, 'policy':
'((ONE or THREE) and (TWO or FOUR))', 'attributes': ['ONE', 'THREE', 'TWO', 'FOUR']}, 'c2': {'alg': 'HMAC_SHA2', 'msg': '{"ALG": 0, "MODE": 2, "IV": "3xchsyuYidxRTKE/wdWcXg==", "CipherText": "1dKUFnI/iAQ
G0lqe16ZNYzxBJP+25Fuy3YfZczfJbBYs2GDnvSQV6s3M4ofIVG8II979QUnRR2ySRl6PfoALIIvS3XdMzuJGJv5TWxhVGzB/Ux7B9XfgTPkz60K71eUL4g8FvOr2CHsii1EaIXtIFdIdOTeEmWAjY1ZjilS8yuVXm3G/zsYDqoaevtaWqtu7YbZ5p06Y0be1qIA0YC1DHL
F4vTIoPgXM6NqzSYNNWW4rR6+Ui+5DA0kirClviJll/ySZiVsOoN+C12LsMIATvgYt40CMKyoMfKvxrECAp9o832fLc7bqQtpvBNQRbdAyHGoVze6sfsLaeXIlAJlDGn/wVtgq7+ls+Sto8MDooOtlvG2aqjsot0Tnll7VYbaktmrnxwllVsq2xtenx8P6RE5SdQOGXFRR6
7UMGEtPhSt0EVraXFWMlWlfqctOM9tu4J1cz/CI72PSBrwowrme8O3Kf10hu3F810vBY0etu6Qrs3zAdCn4tyY2Jy/seL0dLoNgarJtCuB2CE+V2+RRQYPM+PltoYuPuhFGpBeWQgKUxVhO1OyVQrMGVjOKLMfyNL4RF+K9uNF1NT84amsD/UUuGBdEKuOU7NeMP4Ki9s/r
ZErgkDHLkTsnzcw0Qe+LTXue4AjGdAIYgldx4dVK+TCch/qAychMddJ+WiBE9xeh89Xg1cnlqAVA5JPJnPbfFlmP5qwwZ5tg+8DtxhlUvjqldFMaEvIqcW82ytuAVoCcccibU+iuA16D7m3KG5koVJALZ+i89Wsf9chUuxWAbP3ysyiA8y8SZ7s8MOpfS1/uBQA4k0jMEHG
vI4m2a6Tk3ALuTBCrZZFt/hemt2/mSn/wfbsQC4A+Hb9Ps7fn/GY+P8hdpJfOO+HeYxyN4xrY5EJACY0QnvbhYQRJaEFlExV5BuiNSS6SZoMAqhq+gzdCWxqjUuYJbwztAkMfrewAvib157VVOXlDKC4fYWjHKRHOtWA3gaCnMhHbTPyfiUV2upIITCSkn73bX1y11a8CwJ
sv4isgHASqjehFieb8F0zf1mzC3TJXiQzCCzLHVd4bhgVe6IhSCRzoGVqNJaBGsHWj/S7FqX+zcSDE3fUsc/MprEGgjqJM5N7HWmvyV6icjATQw4d0+OzzM73GNMvMCnIttFPIaG8S8w8XmNzT2YNN1ZVusjVm/rfFafvDnHftOFCY10GFswUSGFAYZyTwPkZyxMGWmgzZr
UaalNxz0g=="}', 'digest': '20776da162fb1e8255c56dab07f1c1c33bc776ad7ac60c7c7da14468441e4f95'}}

ContentID

<Response [200]>
Prescription.txt: Qme7Sq8gLmE875kE79QyWWFy9wqQ4yHnTEHMur511PrZfF
Folder CID: QmWP13wr64fit1Nt7PUpM3wxBro5x1LvjzWgiZFNXtJFrH

DecryptedData

[20555182183685353122571563530325425353938068740530724862682245180051174551690462118295274887059378445974567978529897865903748426832116574730356637778792 71, 3720114716169197903951888851439982024564117839
55350922086643760983683265274086684729437984150118125558538645197435024675470140294910571580335323873915228 80]
b'John Doe\xe2\x80\x99s Bags of Medications\n(Note: you would only know what these are if you accessed an electronic pill identifier site like\nDrugs.com)\nMorning Ziplock:\n\xe2\x80\xa2 Allopurinol 2 50
 mg tablets: learn he takes 1 or 2 a day depending on whether he has gout\n\xe2\x80\xa2 Aspirin \xc2\xbd tablet: doctor told him to take \xc2\xbd tablet\n\xe2\x80\xa2 Clopidogrel 75 mg tablet\n\xe2\x80\x
a2 Colchicine 0.6 mg tablet\n\xe2\x80\xa2 Glyburide 1.25 mg tablet\n\xe2\x80\xa2 Toprol XL 50 mg tablet\n\xe2\x80\xa2 Amiloride 5 mg tablet\n\xe2\x80\xa2 Enalapril 20 mg tablet\n\xe2\x80\xa2 Tylenol Arth
ritis 2 650 mg tablets\nAfternoon Ziplock:\n\xe2\x80\xa2 Tylenol Arthritis 2 650 mg tablets\nPM Ziplock:\n\xe2\x80\xa2 Colchicine 0.6 mg tablet\n\xe2\x80\xa2 Glyburide 1.25 mg tablet\n\xe2\x80\xa2 Simvas
tatin 80 mg tablet\n\xe2\x80\xa2 Warfarin 5 mg tablet\n\xe2\x80\xa2 Amiloride 5 mg tablet\n\xe2\x80\xa2 Enalapril 20 mg tablet\n\xe2\x80\xa2 Tylenol Arthritis 2 650 mg tablets\nAlso has:\n\xe2\x80\xa2 Ni
troglycerin bottle of 0.4 mg tablets \xe2\x80\x93 takes 1 QD or QOD\n\xe2\x80\xa2 Albuterol inhaler: prn. Does not use often.\n'
osboxes@osboxes:~/Desktop/FortiRX$
```

Fig. 5. FortiRx Implementation Showing Encrypted Prescription and Successful Upload to IPFS

9 Experimental Results

This section discusses the results from the implemented proposed FortiRx application. Along with that security, timing, and cost analysis are also performed to evaluate the proposed system's reliability and adaptability.

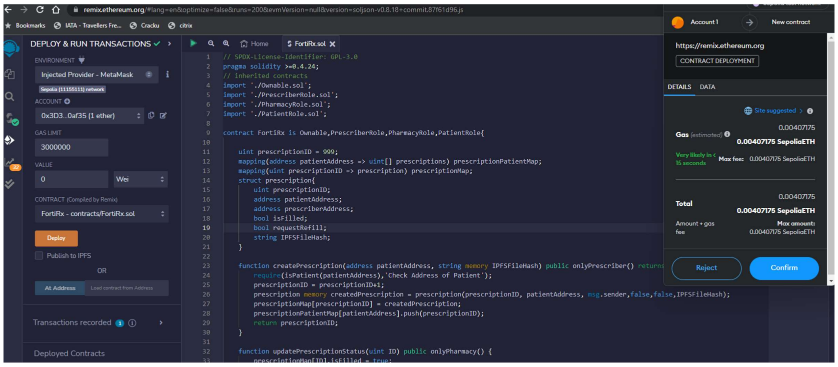
9.1 Results

The first step of the proposed FortiRx application is creating a prescription text document. A sample electronic prescription is taken for testing purposes and is copied into a text file. The file is then encrypted using the CP-ABE scheme with a pre-defined access policy. Encrypted files are then uploaded to IPFS using infra. During the retrieval process, based on the CID the prescription data is retrieved and decrypted using a secret key generated from the attributes assigned to an entity. Encrypted prescription, Content ID, and Decryption of the prescription data can be seen in Figure 5.

Remix IDE is used to deploy smart contracts in sepolia testnet. MetaMask digital wallet is used for maintaining user accounts and sending transactions to Ethereum. Transaction for deploying the FortiRx contract is shown in Figure 6. Once the contract is deployed Different accounts are assigned different roles: Prescriber (Physician), Pharmacy, and Patient accounts. Some of the important transactions and their corresponding addresses and hashes are shown in Table 2.

9.2 Validation

Security Analysis Security analysis is performed on the proposed FortiRx to check the feasibility and adaptability of the system in real-world E-Prescription systems.
Threat 1: An adversary trying to gain access to sensitive information of patient



remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js

DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: Injected Provider - MetaMask

Sepolia (11155111) network

ACCOUNT: 0x3D3...0af35 (1 ether)

GAS LIMIT: 3000000

VALUE: 0 Wei

CONTRACT (Compiled by Remix): FortiRx - contracts/FortiRx.sol

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.4.24;
// inherited contracts
import './Ownable.sol';
import './PrescriberRole.sol';
import './PharmacyRole.sol';
import './PatientRole.sol';

contract FortiRx is Ownable,PrescriberRole,PharmacyRole,PatientRole{

    uint prescriptionID = 999;
    mapping(address patientAddress => uint[] prescriptions) prescriptionPatientMap;
    mapping(uint prescriptionID => prescription) prescriptionMap;
    struct prescription{
        uint prescriptionID;
        address patientAddress;
        address prescriberAddress;
        bool isFilled;
        bool requestRefill;
        string IPFSFileHash;
    }

    function createPrescription(address patientAddress, string memory IPFSFileHash) public onlyPrescriber() returns
        require(isPatient(patientAddress),'Check Address of Patient');
        prescriptionID = prescriptionID+1;
        prescription memory createdPrescription = prescription(prescriptionID, patientAddress, msg.sender,false,false,IPFSFileHash);
        prescriptionMap[prescriptionID] = createdPrescription;
        prescriptionPatientMap[patientAddress].push(prescriptionID);
        return prescriptionID;
    }

    function updatePrescriptionStatus(uint ID) public onlyPharmacy() {
        prescriptionMap[ID].isFilled = true;
```

Account 1 → New contract

https://remix.ethereum.org

CONTRACT DEPLOYMENT

DETAILS DATA

Site suggested

Gas (estimated): 0.00407175
0.00407175 SepoliaETH

Very likely in < 15 seconds — Max fee: 0.00407175 SepoliaETH

Total: 0.00407175
0.00407175 SepoliaETH

Amount + gas fee — Max amount: 0.00407175 SepoliaETH

Fig. 6. Deployment of Smart Contract in Sepolia Testnet.

Table 2. Role Assigned Accounts and Transaction Hashes on Sepolia Testnet

Feature	Value
Physician Account Address	0x3d352313f4f5561d0ffbfda205b52a3c3b70af35
Pharmacy Account Address	0x3D352313F4f5561D0fFBfda205B52A3c3b70af35
Patient Account Address	0x2a9884dfa7E6890FE8AA99FE2486c613C32b697a
Contract Deployment Hash	0x798d1f5ff49f9df09b9856db2646cebc2029d5cd2a45c5ef0c1b9 acb9f217c6f
Prescription Content ID	Qme7Sq8gLmE875kE79QyWWFy9wqQ4yHnTEHMur511PrZfF
Prescription Creation Hash	0xda5bd0ce943325696e91bfe140bd8cdd60eafdca6f2a41b0722 1e499bfe7f1f7

Solution: Proposed FortiRx makes use of a robust CP-ABE cryptography scheme. This will ensure the prescription data is only accessible to the authorized entities with assigned attributes and avoid data leakages. As blockchain uses identities based on the PKI system, anonymity is maintained while sending transacting in blockchain thereby preserving data privacy.

Threat 2: Fake transactions generated by an adversary in the blockchain network to introduce falsified information.

Solution: RBAC mechanism based on smart contracts and modifiers is defined in the proposed FortiRx which will ensure the functions are well-defined and given proper access to different entities. An adversary who will not have these privileges will not be able to send the transaction and create falsified information in the network.

Threat 3: Data manipulations by the adversary or prescription abuse by duplication.

Solution: Blockchain creates an immutable ledger with all the transactions generated from each and every participating entity. This ledger is copied at each node and acts as a single source of truth which is difficult to manipulate and easy to verify. This ensures no data modifications can be done to the data stored on the blockchain.

Timing and Cost Analysis To perform timing and cost analysis, each smart contract interaction is repeated 10 times, and the average time taken for confirmation of transaction and gas cost is evaluated. Results from the analysis can be seen in Fig. 7 and Fig. 8. From timing analysis, it can be seen transaction times are not changed much based on the type of function as it mainly depends on the network congestion at the time of the transaction. So average delays ranging from 12-18 Seconds is an acceptable delays in real-world applications. Cost analysis shows a significant difference between contract deployment and other functions as the cost depends on the number of instructions that need to be executed by the Ethereum Virtual Machine (EVM). The cost of deployment of a smart contract is 7.2$ converting ETH to US at the conversion rate of 1 ETH = 18000 USD whereas for all other functions is nearly a dime. The cost and latency can be avoided if private blockchain is implemented instead of using public blockchain.

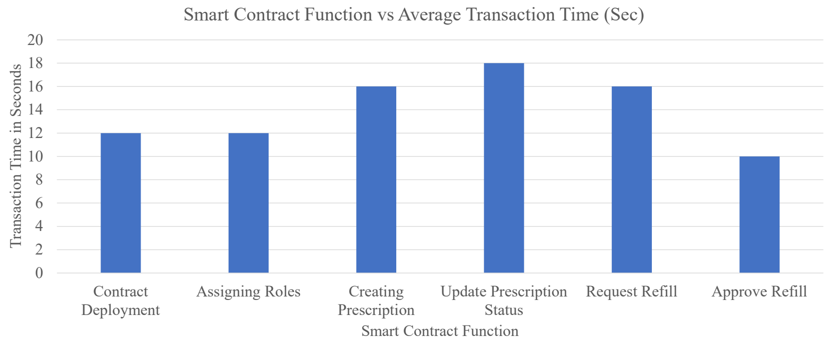


Fig. 7. Average Transaction Times of Smart Contract Functions.

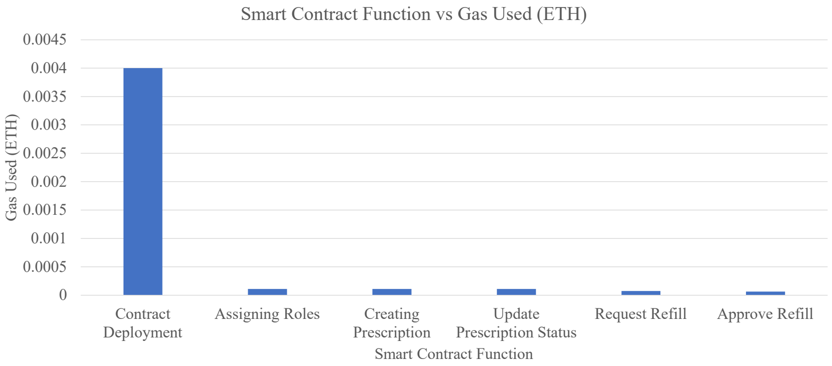


Fig. 8. Average Transaction Cost of Smart Contract Functions.

10 Conclusions and Future Research

In this work, we have proposed a novel idea of an E-Prescription system implemented using blockchain and smart contracts. The cost overhead of large on-chain data is addressed in the proposed FortiRx by implementing distributed data storage using IPFS. A robust access control mechanism using CP-ABE which ensures the efficient sharing of data between a dynamic group of data users has been implemented for preserving the privacy of patient information. This patient-centric approach will give more control to the data owners and efficiently manage the access mechanisms. Proof-of-Concept of proposed FortiRx is implemented and analyzed for scalability and reliability in real-world scenarios. Results from the analysis have shown the robustness of the proposed system for different security threats.

In future work, more scenarios and interactions will be included in the smart contract business logic to build a complete solution for the E-Prescription system. Implemented CP-ABE, even though provides a robust access control mechanism, it needs a centralized trusted entity to distribute attributes to the participants. A decentralized trustless key distribution mechanism can be beneficial in this case. Future work will be in the above scenarios along with providing a user-friendly interface for easy interaction with the system.

References

1. Ababneh, M.A., Al-Azzam, S.I., Alzoubi, K.H., Rababa'h, A.M.: Medication errors in outpatient pharmacies: comparison of an electronic and a paper-based prescription system. Journal of Pharmaceutical Health Services Research **11**(3), 245–248 (apr 2020). https://doi.org/10.1111/jphs.12356
2. Ahmadi, V., Benjelloun, S., Kik, M.E., Sharma, T., Chi, H., Zhou, W.: Drug governance: IoT-based blockchain implementation in the pharmaceutical supply chain. In: Proc. Sixth International Conference on Mobile And Secure Services (MobiSecServ) (2020). https://doi.org/10.1109/mobisecserv48690.2020.9042950
3. Alnuaimi, A., Alshehhi, A., Salah, K., Jayaraman, R., Omar, I.A., Battah, A.: Blockchain-based processing of health insurance claims for prescription drugs. IEEE Access **10**, 118093–118107 (2022). https://doi.org/10.1109/access.2022.3219837
4. Bapatla, A.K., Mohanty, S.P., Kougianos, E., Puthal, D., Bapatla, A.: PharmaChain: A blockchain to ensure counterfeit-free pharmaceutical supply chain. IET Networks **12**(2), 53–76 (jul 2022). https://doi.org/10.1049/ntw2.12041
5. Bapatla, A.K., Mohanty, S.P., Kougianos, E., Puthal, D.: PharmaChain 2.0: A blockchain framework for secure remote monitoring of drug environmental parameters in pharmaceutical cold supply chain. In: Proc. IEEE International Symposium on Smart Electronic Systems (iSES) (2022). https://doi.org/10.1109/ises54909.2022.00046
6. Bapatla, A.K., Mohanty, S.P., Kougianos, E., Puthal, D.: PharmaChain 3.0: Blockchain integrated efficient QR code mechanism for pharmaceutical supply chain. In: Proc. OITS International Conference on Information Technology (OCIT) (2022). https://doi.org/10.1109/ocit56763.2022.00121
7. for Disease control, C., (CDC), P.: Health insurance portability and accountability act of 1996 (hipaa), `https://bitcoin.org/bitcoin.pdf`, last Accessed: 2023-03-18
8. Food, Administration, D.: Working to reduce medication errors (2019), `https://www.fda.gov/drugs/information-consumers-and-patients-drugs/working-reduce-medication-errors`, last Accessed: 2023-03-18
9. Guo, H., Li, W., Nejad, M., Shen, C.C.: Access control for electronic health records with hybrid blockchain-edge architecture. In: Proc. IEEE International Conference on Blockchain (Blockchain) (2019). https://doi.org/10.1109/blockchain.2019.00015

10. Huang, J., Qi, Y.W., Asghar, M.R., Meads, A., Tu, Y.C.: MedBloc: A blockchain-based secure EHR system for sharing and accessing medical data. In: Proc. 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE) (2019). https://doi.org/10.1109/trustcom/bigdatase.2019.00085
11. Ionescu, S.V.: E-prescription using blockchain technology. In: Proc. IEEE International Conference on Blockchain, Smart Healthcare and Emerging Technologies (SmartBlock4Health) (2022). https://doi.org/10.1109/smartblock4health56071.2022.10034520
12. Lamport, L., , Shostak, R., Pease, M., and: The byzantine generals problem. In: Concurrency: the Works of Leslie Lamport. Association for Computing Machinery (oct 2019). https://doi.org/10.1145/3335772.3335936
13. Meyliana, Surjandy, Fernando, E., Cassandra, C., Marjuki: Propose model blockchain technology based good manufacturing practice model of pharmacy industry in indonesia. In: Proc. 2nd International Conference on Innovative and Creative Information Technology (ICITech) (2021). https://doi.org/10.1109/icitech50181.2021.9590120
14. Musamih, A., Jayaraman, R., Salah, K., Hasan, H.R., Yaqoob, I., Al-Hammadi, Y.: Blockchain-based solution for the administration of controlled medication. IEEE Access **9**, 145397–145414 (2021). https://doi.org/10.1109/access.2021.3121545
15. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system, `https://bitcoin.org/bitcoin.pdf`, last Accessed: 2023-03-18
16. Organization, W.H.: Medication without harm, `https://www.who.int/initiatives/medication-without-harm`, last Accessed: 2023-03-18
17. Puthal, D., Mohanty, S.P., Kougianos, E., Das, G.: When do we need the blockchain? IEEE Consumer Electronics Magazine **10**(2), 53–56 (mar 2021). https://doi.org/10.1109/mce.2020.3015606
18. Shahnaz, A., Qamar, U., Khalid, A.: Using blockchain for electronic health records. IEEE Access **7**, 147782–147795 (2019). https://doi.org/10.1109/access.2019.2946373
19. Taylor, A., Kugler, A., Marella, P.B., Dagher, G.G.: VigilRx: A scalable and interoperable prescription management system using blockchain. IEEE Access **10**, 25973–25986 (2022). https://doi.org/10.1109/access.2022.3156015
20. Thatcher, C., Acharya, S.: Pharmaceutical uses of blockchain technology. In: Proc. IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS) (2018). https://doi.org/10.1109/ants.2018.8710154
21. Velo, G.P., Minuz, P.: Medication errors: prescribing faults and prescription errors. British Journal of Clinical Pharmacology **67**(6), 624–628 (jun 2009). https://doi.org/10.1111/j.1365-2125.2009.03425.x
22. Vora, J., Nayyar, A., Tanwar, S., Tyagi, S., Kumar, N., Obaidat, M.S., Rodrigues, J.J.P.C.: BHEEM: A blockchain-based framework for securing electronic health records. In: Proc. IEEE Globecom Workshops (GC Wkshps) (2018). https://doi.org/10.1109/glocomw.2018.8644088
23. Waters, B.: Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: Public Key Cryptography – PKC 2011, pp. 53–70. Springer Berlin Heidelberg (2011). https://doi.org/10.1007/978-3-642-19379-8_4
24. Wolf, M.S., Davis, T.C., Tilson, H.H., Bass, P.F., Parker, R.M.: Misunderstanding of prescription drug warning labels among patients with low literacy. American Journal of Health-System Pharmacy **63**(11), 1048–1055 (jun 2006). https://doi.org/10.2146/ajhp050469
25. Ying, B., Sun, W., Mohsen, N.R., Nayak, A.: A secure blockchain-based prescription drug supply in health-care systems. In: Proc. International Conference on Smart Applications, Communications and Networking (SmartNets) (2019). https://doi.org/10.1109/smartnets48225.2019.9069798
26. Zaghloul, E., Li, T., Ren, J.: Security and privacy of electronic health records: Decentralized and hierarchical data sharing using smart contracts. In: Proc. International Conference on Computing, Networking and Communications (ICNC) (2019). https://doi.org/10.1109/iccnc.2019.8685552