Blockchain-Based Processing of Health Insurance Claims for Prescription Drugs

AYSHA ALNUAIMI1, AMNA ALSHEHHI2, KHALED SALAH2, (Senior Member, IEEE), RAJA JAYARAMAN1, ILHAAM A. OMAR1, AND AMMAR BATTAH1

1Department of Industrial and Systems Engineering, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates
2Department of Electrical Engineering and Computer Science, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates

Corresponding author: Raja Jayaraman (raja.jayaraman@ku.ac.ae)

This work was supported by the Khalifa University of Science and Technology under Award CIRA-2019-001.

ABSTRACT The current legacy system used in processing health insurance claims causes a huge amount of financial loss every year due to fraud claims. It is also highly prone to privacy and security threats due to the use of traditional methods. Health insurance claims for prescription drugs are one such claim that is highly prone to being tampered with. Also, there is a lack of linkage in the prescription data between the medical care provider and the pharmacy, which also leads to miscommunication and the use of false prescriptions. In this paper, we propose processing health insurance claims for drug prescriptions using blockchain technology that manages the processing of prescription drugs in a private, secure, trustworthy, and decentralized manner. The proposed system utilizes a private Ethereum blockchain. The system includes two smart contracts: registration and approval smart contracts, which provide traceability and trustworthiness to the system. The system was integrated with off-chain storage (IPFS) and a fronted decentralized applications (DApps), which improves the accessibility of centralized patient information by authorized parties. To maintain the privacy of the data, a gateway is used to filter the data that can be viewed by the participants. We present the system architecture, sequence diagrams, entity-relationship diagram, and algorithms to demonstrate the working principles of the proposed process for processing health insurance claims for prescription drugs using blockchain. The performance of the proposed solution is evaluated by conducting security analysis and comparing it to the existing solutions. Our smart contract code is publicly available on GitHub.

INDEX TERMS Blockchain, ethereum, insurance claims, prescription drug, smart contracts.

I. INTRODUCTION
The health insurance sector has been increasingly studied to improve the current crumbling traditional system. A health insurance claim is a bill that represents a transaction between a medical service provider and an insurance company to cover the expenses of a patient obtaining a service, such as medical, pharmaceutical, and dental services.

There are two main types of health insurance claims: cashless and reimbursement. In the reimbursement type, the patient pays out of pocket for all services rendered by the physician or hospital, and the insurance provider reimburses. In cashless claims, the insurer directly covers all hospitalization expenses for the hospital. To receive the benefit of a cashless claim, an insured must be hospitalized exclusively at a network hospital approved by the insurance provider. Cashless claims cover both planned and unplanned hospitalizations. In this research, we are considering cashless as it is the most popular type of claim universally [1].

Figure 1 shows a typical process flow of how health insurance claims for prescription drugs are made. It is started when a patient approaches a network hospital for cashless treatment. Then the registration begins, where the patient provides personal details and insurance information to the healthcare provider. After the registration, the care provider validates the patient’s details and insurance information, and then the patient is referred to a physician to conduct the necessary diagnosis. Based on the diagnostic results, the physician writes his medical codes and creates a prescription which is stored in the healthcare centralized system.
A. Alnuaimi et al.: Blockchain-Based Processing of Health Insurance Claims for Prescription Drugs

FIGURE 1. Typical process flow for processing health insurance claim for prescription drug.

pharmacy is notified when the prescription is uploaded to the centralized system. The pharmacist then contacts the health insurance company for prescription approval. The insurance company verifies the claim with their policy benefits to evaluate coverage and eligibility. In this step, they verify whether the patient has a deductible, accumulated co-pay, or out-of-pocket expenses, and they decide whether the medical claim is valid and how much of the claim they will pay [2]. There are two scenarios based on the insurance company’s decision. The first one is an “approved claim” in which the pharmacist notifies the patient to collect medications and the claim is paid after the drugs are received. On the other hand, the claim may be rejected, and the pharmacist will ask if the patient is willing to pay out of pocket or to cancel the prescription [3], [4]. Recently, it came to the realization that this system is struggling with many drawbacks, such as claim abnormality, security and privacy issues, time and cost consumption [5].

The most significant disadvantage is claim abnormality, which can be divided into three categories: fraud, abuse, and errors. A fraud claim is one that is purposefully deceived, misrepresented, or concealed in order to obtain payment from the insurance company. Abuse claims result in unnecessary costs through the use of exorbitant or inappropriate medical services that do not align with acceptable business or medical practice. Errors are unintended mistakes made by one or some of the parties that cause improper payment [5]. These abnormalities cause major losses of money to governments and insurance companies across the globe. A recent study suggests that fraud claims waste 15% of the total claims in the insurance sector [6]. In the US, the Department of Health and Human Services Centers estimates healthcare fraud in 2010 to be between $77 billion and $259 billion [7]. In the UK, it is reported that fraud insurance claims result in an annual loss of over one billion pounds. According to another report, the Indian industry loses between 6 and 8 billion INR per year as a result of counterfeit claims [6]. The main cause of all this money loss is that the current system is built on trust. The insurance company trusts the information provided by the patient and the service provider [7]. In 2020 in the US, 345 defendants were charged with fraud, which involved over 100 doctors, nurses, and medical professionals. Many laws and regulations have been enacted to address this issue, but it remains a major challenge [8].

Another issue is privacy and security. The insurance claim, as stated before, includes information about the patients, which causes a major concern about the security of their information. In the US, 50% of people are concerned about their medical data since they believe the healthcare system is not properly prepared for cyberattacks. Furthermore, healthcare organizations ranked the lowest in cybersecurity among seven other industries globally. It failed in almost all components, such as security in the cloud environment, data center, desktops, laptops, mobile devices, and others [7]. Also, it has been reported by HIPPA journal that in the first quarter of 2018, there have been over 77 breaches of healthcare data, which caused data leakage of more than a million records. Although the number of attacks is decreasing, the severity of them is worsening. The centralization of healthcare data continues to be a major threat [9]. The last challenge is the time and cost of consumption. This traditional manual paperwork process takes time, effort, and money. It is estimated that $375 billion is wasted by insurance companies in paperwork [7].

Therefore, a novel system is highly required to overcome the major baggage that is caused by the current insurance
claim system. A blockchain technology-based system is a strong candidate that will resolve the issues of security, trust, and transparency due to its features like immutability, distribution, traceability, and decentralization. A blockchain system validates and stores transactions in a decentralized ledger. Data integrity is recognized as a key security challenge that affects both data transmission and data collection. To ensure data integrity, blockchain technology offers three essential conditions: immutability, transparency and traceability of transactions. This ensures that data committed to the blockchain is immutable preserved even after moving to new state in the ledger. As such, an audit trail that can be used to ensure the validity and integrity of data. This is achieved by the underlying hash mechanism, where a hash pointer points to the previous block in the blockchain. If a block of data is tampered with the hash would change, which would in turn change the hash of all the proceeding blocks as each block contains the hash of the previous block [20]. To the best of our knowledge, there is a lack of work on smart systems for medical prescription processing, especially blockchain-based solutions. This paper proposes a blockchain-based insurance claim for prescription drug processing systems to eliminate fraud cases. The main contributions of this paper are as follows:

- We propose a private Ethereum blockchain-based solution to automate health insurance claim processes for prescription drugs in a fully decentralized, traceable, transparent, private, secure, and trustworthy manner.
- We present detailed algorithms depicting the interactions among stakeholders including sequence diagrams that explain the proposed solution for processing health insurance claims.
- We develop two smart contracts in regards to health insurance claims for prescription drugs.
- We evaluate our proposed model and our developed smart contracts using security analysis to demonstrate the robustness of our proposed solution against major security vulnerabilities.
- We compare our proposed approach with other solutions demonstrating the advantages and feasibility of our approach.

The rest of the paper is structured as follows. Sections II and III present the background and related work, which is classified into two categories, namely, blockchain and non-blockchain-based health insurance claim solutions. Section IV describes the proposed blockchain-based solution for health insurance claims for prescription drugs. Sections V and VI present the implementation, testing, and validation details of the proposed solution. In Section VII, we conduct the security analysis of the proposed approach. Section VIII is a summary of our results and concluding remarks.

II. BACKGROUND

In this section, we briefly introduce blockchain technology and expand the explanation of Ethereum.
2) THE DATA LAYER
A transaction is the commerce between a sender (EOA) and a recipient (another EOA or contract account). Every transaction has:

- Nonce: A counter to track the number of transactions initiated by the sender
- Recipient: Indicates the transaction’s destination (contract account of EOA)
- Value: The amount of currency in Wei to be sent from the sender to the recipient
- Input: The transaction-related data
- Gas Price and Gas Limit: The unit price and maximum amount of gas that the sender is willing to pay to the winning miner

\((v, r, s)\): the Elliptic Curve Digital Signature Algorithm (ECDSA) signature of the sender

Once a transaction is executed, the states of the accounts involved are updated, and hence the blockchain is updated. The data structure for storing Ethereum blockchain data is known as a trie. It stores (key, value) pairs. A key is the path from the root to a leaf node while the value is contained within the leaf node. A block header could point to a state trie, a transaction trie, and a receipt trie. Each contract account uses a separate storage trie to bookkeep the persistent data of the contract. Since the state of the blockchain dynamically changes, Ethereum has one state trie.

3) THE CONSENSUS LAYER
This layer regulates the generation and verification of a block. This is done by enforcing network rules for the nodes to reach consensus about the broadcast transaction. There are two main consensus protocols: proof-of-work and proof-of-stake. Currently, Ethereum uses the GHOST consensus protocol, which is a proof-of-work protocol run by miners who compete to solve the mathematical puzzle and create new blocks with successfully processed transactions to link them with current blocks. The winner shares the block with the network and earns ethers. According to this protocol, the main chain is the chain with the heaviest branch. It is the branch with the highest cumulative block difficulty.

4) THE NETWORK LAYER
Ethereum has a peer-to-peer network. Each peer (node) shares and stores information about the entire network. For node discovery and routing purposes, each node maintains a dynamic routing table of 160 buckets, and each bucket contains up to 16 entries of other nodes’ IDs, IP addresses, and UDP/TCP ports. To find target clients, Ethereum uses the RLPx protocol, and it uses the Ethereum Wire Protocol to facilitate the exchange of information between clients.

5) THE ENVIRONMENT
The Ethereum blockchain operates in a four-layer environment: a web interface for clients to use the Ethereum...
blockchain; a database to store all the clients’ blockchain-related data; cryptographic mechanisms for security reasons; and the Internet infrastructure to facilitate blockchain communications between peers [10].

III. RELATED WORK
In this section, we discuss the existing solutions that have been proposed to address the important challenges in processing health insurance claims. We considered both blockchain and non-blockchain-based approaches.

A. NON-BLOCKCHAIN-BASED SOLUTIONS
In the non-blockchain section, we will highlight various approaches that were conducted to enhance the processing of health insurance claims. One of these approaches is the improvement of decision tree classifier accuracy for healthcare insurance fraud prediction by using the Extreme Gradient Boosting algorithm. In [11], the authors intend to examine statistical modeling methods that use cutting-edge technology to evaluate bogus health benefits. The proposed decision tree algorithm method does not require normalization and thoroughly analyzes each branch to identify decision nodes that require further investigation. Random forest and extreme gradient boosting are included in the algorithm. A decision tree-based classification learning algorithm uses a gradient boosting framework to improve weak learners and perform cache hardware optimization by reserving an internal buffer for each thread to save statistics gradient. However, the extreme gradient boosting is hard to tune and almost impossible to scale up. In addition, it lacks scalability due to its slowness.

In [12], the authors combined graph mining, social network analysis, and data visualization. This combination aided in the transformation of a visual pattern discovered during the data exploration phase into a network model that reveals underlying mutual referrals from the claims database, allowing for an in-depth, scalable analysis of such relationships. They examined a dataset containing only claims relating to physician-performed procedures. As a result, the claims data can be mapped into a social network by considering connections between healthcare professionals or patients. They demonstrated how information visualization and social networking techniques can be used to analyze health insurance claims data, primarily by mapping physicians using shared patients as a proxy. However, this technique of social networking lacks immutability.

B. BLOCKCHAIN-BASED SOLUTIONS
In [13], MIStore system has been proposed. It is a blockchain-based health insurance storage system that operates on the Ethereum platform. In a basic instance of the system, there is a hospital, a patient, an insurance company, and n servers. The hospital implements a threshold (t, n) MIStore protocol between n servers. Any node may become some hospital’s server if both the node and the hospital agree. Moreover, the hospital stores patient consumption data on the blockchain and protects it with n servers. Any t-server can assist insurance companies in obtaining the sum of a subset of patient cost data, and the server can perform homomorphic calculations on this data. However, if n-t or more servers are truthful, n servers will be unable to learn anything from the patient’s expenditure data recorded on the blockchain. Furthermore, they proposed that MIStore use Practical Byzantine Fault Tolerance (PBFT) as the blockchain consensus scheme, with all relevant data stored on the blockchain. The blockchain platform severely limits MIStore’s efficiency.

The authors in [14] aim to build a structure for an efficient way to handle security-related exchanges that relies on the blockchain-driven phase. They developed various sponsor-based smart contracts for various policies. A Hyperledger fabric is used as the blockchain framework in this case. The proposed system is primarily intended to eliminate fraud and risks in the existing system by employing the concept of a proof-of-work algorithm in order to make the system more secure. The proposed blockchain distribution platform’s development goal is to execute insurance claims from insurance companies as smart contracts and update the results in the system. The proposal is divided into three modules: registration, treatment, and claims. Insurance claims are divided into three categories: stakeholders, patients, and hospitals.

Moreover, in [9] the authors considered two of the most common fraud scenarios: patient theft of limited health insurance benefits and the use of multiple insurance policies for additional benefits. They defined a conceptual model in which misleading information can be eliminated in accordance with the HIPAA privacy rules established by the United States Department of Health and Human Services. They use BigchainDB technology, which provides the blockchain’s public functions.

The above-mentioned blockchain-based contributions explain how blockchain can enhance the current health insurance system and eliminate several frauds. However, in some of these papers, the smart contracts are not fully investigated. In addition, the testing and results of the proposed model are not fully described. In other words, the full implementation of the solution was not established.

IV. BLOCKCHAIN-BASED SOLUTION FOR PROCESSING HEALTH INSURANCE CLAIM OF PRESCRIPTION DRUG
In this section, we introduce a blockchain-based solution to process health insurance claims for prescription drugs in a tamper-proof, secure, private, confidential, and trustable manner. The system is built on a private Ethereum blockchain to ensure confidentiality and privacy of records as it contains sensitive information about patients. Therefore, a permissioned blockchain can be viewed only by authorized entities.

Figure 2 illustrates the high-level system architecture of the proposed blockchain-based approach. As it is shown in the figure, there are two smart contracts: the registration smart contract and the approval smart contract. These smart contacts can be accessed by authorized actors through Fronted
Decentralized Applications (DApps). In addition, the system architecture includes an application programming interface (API), such as JSON RPC, Web3 and Infura. APIs can be used as linkers between smart contracts and the software devices that access them (Dapps). Moreover, there is an off-chain storage system, IPFS, that is used to store large-sized files. Also, in order to maintain privacy within the network, a gateway has been added to filter what can be accessed by patients. In other words, patients will be able to see their own data to maintain other patients’ privacy. The components of the proposed solution are described below:

- **Actors:** The main stakeholders of the system are the regulatory authority, patient, physician, pharmacies, and insurance company. Each entity will play a unique role in processing health insurance claims for prescription drugs, granting them access to restricted functions in the smart contracts. Also, they will have access to the information that is stored on chain and off chain.

- **Decentralized Storage Systems:** We utilize a distributed peer-to-peer file system to enhance the storage capabilities of the system and increase its scalability. In specific, we focus on Interplanetary File System (IPFS) which is one of the most common distributed P2P file systems. The file system stores data persistently, as it is distributed among several participants. Furthermore, IPFS storage is immutable as the data is
content-addressable, meaning that a change in content changes the corresponding address. The content on IPFS pointed to using a content identifier (CID), which is a cryptographic hash of the data [19]. This complements the nature of the blockchain, as such, a CID is usually submitted on-chain while large files are stored on the IPFS. Since both the blocks and the link to the content are immutable, integrity is guaranteed as well as authenticity since transactions are signed by participants. Furthermore, the IPFS can store large volumes of data as the network grows, unlike traditional storage systems that are limited by owned physical assets. As such, both the blockchain and IPFS work harmoniously to maintain the required security features as well as meeting scalability needs, by relieving the blockchain from processing and storing large amounts of data. Participants of the systems would simply access the data on IPFS through the index on the blockchain, knowing that the data is tamper-free. Since IPFS is public P2P system aimed to resist censorship and tampering, securing data would need to be implemented on top of it. The owner of the data would encrypt the data using a symmetric key as it is fast and proceeds to encrypt the symmetric key using the recipients public key. As such, we take advantage of the light symmetric encryption while still providing the asymmetric security properties. The recipient can simply proceed to download the data and decrypt it using their private key. Such an approach has been discussed in a previous work [22].

• **Ethereum Distributed Ledger:** The Ethereum distributed ledger is the main component of the system.
It contains all transactions, logs, and events that are recorded in a tamper-proof manner. Thus, that ledger ensures traceability, transparency, and accountability of health insurance claims. Moreover, the private Ethereum blockchain adds privacy, confidentiality, and authorization.

- **Ethereum Smart Contracts**: There are two smart contacts in the system: registration and approval. The registration smart contract is responsible for giving permission to the system’s stakeholders, where they are registered by regulatory authority. The second smart contact, approval, is responsible for processing the claim of a prescription drug, starting from creating the prescription until the claim is paid by the insurance company.

- **Blockchain Clients**: The blockchain client is used as an access point between the Ethereum blockchain and the front-end DApp, which enables the latter to fetch events and logs from the blockchain and display them to the user/patient. Another role of the blockchain client is ensuring that only authorized nodes participate in their assigned network and validate transactions, and this ensures that unauthorized entities have no access to the logs and history of the private transactions. Each entity requires a client and it enables them to run consensus across the participants and set the details of encrypted communication, open-source examples of such clients are Besu⁴ and GoQorum.² Clients can also have multi-tenancy where an establishment has its own client to manage their data. As such, they are able to filter who is able to access shared information through the help of Private Transaction Managers (PVMs). For instance, a doctor has access to medical records to several patients from the hospital database, but a patient is only given access to their own medical records. Therefore, selective access to data is achieved through utilization of blockchain clients.

**A. SEQUENCE DIAGRAM OF THE PROPOSED BLOCKCHAIN-BASED SOLUTIONS**

The interactions among the stakeholders, smart contracts, and storage systems are represented in this subsection.

Figure 3 demonstrates the sequence diagram of all interactions in the proposed system. The sequence starts with deploying a registration smart contract where a regulatory authority is responsible for registering all entities. Then, the approval smart contract is responsible for giving permission to the system’s stakeholders, where they are registered by regulatory authority. The second smart contact, approval, is responsible for processing the claim of a prescription drug, starting from creating the prescription until the claim is paid by the insurance company in order to process the prescription, and an event will be emitted upon that request. The insurance company will evaluate the approval request based on their policies and the patient’s plan and coverage. If the request is approved, the pharmacy will prepare the medicines for collection and an event will be sent. The patient will collect the medication and an event will be emitted. Once the medication is collected, the pharmacy will send a claim to be paid by the insurance company. On the other hand, if the request is rejected, an event of the rejection will be emitted.

**B. PERMISSIONED BLOCKCHAIN**

In a sensitive domain such as healthcare, exposure, tampering or loss of data can be detrimental. As such, we proposed a private permissioned Ethereum blockchain with the focus on achieving privacy, confidentiality and access control. Unlike the public Ethereum mainnet, deployers get to build the private blockchain using Ethereum clients such as Hyperledger Besu and GoQorum. We do not confine our solution to a specific implementation of the private blockchain as to not restrict the offered customizability, we utilize concepts common to private blockchains. Authenticity of transactions is maintained similarly between public and private where a transaction is signed using the public key of the sender. In a permissioned chain, users are authenticated by a trusted authority through their submitted digital certificate. Ethereum clients are paired with a PVM, commonly Tessera,³ which offers flexible privacy of transactions on the network. The PVM allows the sender to specify privacy groups or send directly to a single receiver. It extends to what transactions participants can read or save, and what nodes they are allowed to connect to. Since private transactions are encrypted, any associated data is kept confidential, which is desired for sensitive information such as the patient data. This extends to data stored off-chain which is encrypted on the distributed storage. Furthermore, private blockchains can employ several consensus protocols, most notably proof-of-authority (PoA) protocols which can be efficiently employed since stakeholders are authenticated and can be assigned their appropriate authority.⁴ Moreover, the authenticated participants are given different authorization levels based on their role. This is achieved by the smart contract which restricts the ability to execute functions based on authority of the caller. The aforementioned concepts facilitate trust within the consortium, maintaining transparency and privacy as required. Our

---

¹https://besu.hyperledger.org/en/stable
²https://docs.tessera.consensys.net/en/stable
³https://docs.tessera.consensys.net/en/stable
⁴https://besu.hyperledger.org/en/stable/private-networks/how-to/configure/consensus/
solution makes use of both the intrinsic blockchain characteristics and the customizable permissioned private blockchain features.

V. IMPLEMENTATION OF THE PROPOSED MODEL
In this section, we introduce the developed algorithms and implementation details of the proposed blockchain-based solution to process insurance claims for prescription drugs.

A. IMPLEMENTATION DETAILS
The proposed solution is built on a private Ethereum blockchain, so only authorized entities can access and execute functions. Only the assigned regulatory authority is allowed to execute the registration smart contract to register all the different entities that can execute their corresponding functions in the approval smart contract.

The entity-relationship diagram of all attributes and functions of smart contracts and authorized entities is depicted in Figure 4. The registration smart contract has only one regulatory authority, therefore it is declared as an address. While the others are declared as mapped, which means that there are multiple physicians, pharmacies, and insurance companies in the system that can execute the same functions. The regulatory authority deploys and executes the registration functions of all authorized entities. The approved smart contract’s attributes and functions are listed in the diagram. There are different enumerating variables in the approval smart contract, including ApprovalRequestState that contains different states of pharmacy confirmation, such as “Pending” and “Approved”. In addition, InsuranceApprovalStates contains different states of approval requests, such as “Pending”, “Approved” and “Rejected”. Moreover, MedicineCollectionState can be “ReadyForCollection” or “Collected”. Furthermore, the paymentState is either “pending” or “paid”. Finally, as there is only one registration smart contract and multiple approval smart contracts, the relationship is represented as 1:n.

B. ALGORITHMS
The major algorithms are illustrated below to clarify the concept behind the two smart contracts, registration and approval.

The registration smart contract is deployed by the regulatory authority and all authorized entities are registered by running their corresponding functions. The registered entity will have permission to execute their respective functions in the approval smart contract later. Algorithm 1 illustrates the registration phase of. If the caller is the regulatory authority, then the entity’s Ethereum address will be added to the authorized entity and an event will be emitted to declare the details of each entity’s registration process.

Prescription creation is explained in Algorithm 2. To execute the PrescriptionCreation function, the caller should be the physician. The patient ID, Drugs CRN, and IPFS hash will be the inputs. Once the prescription is created, the physician will upload the IPFS hash. Then, an event will be broadcast to declare that the prescription has been created and uploaded.

Algorithm 3 demonstrates the pharmacy selection process. To execute the PharmaciesSelection function, the caller must be the patient and the selected pharmacies should be registered. Moreover, the number of chosen pharmacies by the patient should be ≤ 5. If these conditions are met, then the ApprovalState is shown as “pending” which means that approval is needed to confirm that the pharmacy can process the prescription. The selected pharmacy that is selected by the patient can execute the PharmacyApproval function to update the ApprovalState to be “Approved” and an event will be emitted.

Algorithm 4 illustrates the insurance approval. The selected pharmacy will execute the RequestInsuranceApproval function to request approval from the insurance company in order to process the prescription. If the caller is the selected pharmacy, then the InsuranceApprovalState is shown as “Pending” and an event is broadcast to declare that approval is requested. After that, the InsuranceApproval function is executed if the caller is the insurance company and the InsuranceRequestState is “Pending”. The insurance company can approve or reject the request. If the request is approved, then the InsuranceApprovalState will be updated to be “Approved” and an event will be emitted. If not, the InsuranceApprovalState will be “Rejected” and an event will be emitted.

Medication collection is described in Algorithm 5. In order to execute the function of MedicationPreparation, the caller must be the selected pharmacy and the

---

Algorithm 1: Registration

\[
\text{Input: Entity EA} \\
1 \text{if caller == regulatory\_authority then} \\
2 \quad \text{Add Entity EA to the authorized Entities} \\
3 \quad \text{Emit an event declaring the details of the entity registration process} \\
4 \text{else} \\
5 \quad \text{Reject.} \\
6 \text{// Entity registration is completed}
\]

Algorithm 2: Prescription Creation

\[
\text{Input: PatientID,Drug1CRN,Drug2CRN, Drug3CRN, IPFSHash} \\
1 \text{if (caller == Physician) then} \\
2 \quad \text{Add IPFSHash} \\
3 \quad \text{Emit an event declaring the prescription has been created and uploaded} \\
4 \text{else} \\
5 \quad \text{Reject.} \\
6 \text{// The physician creates and uploads prescription}
\]

Algorithm 4 illustrates the insurance approval. The selected pharmacy will execute the RequestInsuranceApproval function to request approval from the insurance company in order to process the prescription. If the caller is the selected pharmacy, then the InsuranceApprovalState is shown as “Pending” and an event is broadcast to declare that approval is requested. After that, the InsuranceApproval function is executed if the caller is the insurance company and the InsuranceRequestState is “Pending”. The insurance company can approve or reject the request. If the request is approved, then the InsuranceApprovalState will be updated to be “Approved” and an event will be emitted. If not, the InsuranceApprovalState will be “Rejected” and an event will be emitted.
Algorithm 3: Pharmacy Selection

**Input:** Pharmacy Address

1. ApprovalRequestState is a customized enumerate variable that describes different states of the request if (caller == patient) ∧ (selectioncounter[msg.sender] < 5) (registeredPharmacy(pharmacyaddress) == true)

   then
   2. selectioncounter[msg.sender] + = 1
   3. ApprovalState = Pending
   4. Emit an event declaring that pharmacies are selected
   5. else
   6. Reject.

   if (caller ∈ RegisteredPharmacies) ∧ (ApprovalState == Pending) then
   8. for i = 0; i < selectioncounter[patient]; i++ do
   9. if (PharmaciesSelection[patient][i] ∈ RegisteredPharmacies) ∧ (PharmaciesSelection[patient][i] == true)
      then
      10. ApprovalState = Approved
      11. Emit PharmacyEligible
      12. Emit Approved
      end
   13. end
   14. Emit PharmacyIneligible
   15. else
   16. Reject.
   17. end

   // the selected pharmacy confirmed they can process the prescription

InsuranceApprovalState must be “Approved”. If these conditions are met, the pharmacy will prepare the medication for collection, and then the MedicationCollectionState will be “ReadyForCollection” and an event will be emitted to declare that the medication is prepared and ready to be collected. The patient will collect the medication if the MedicationCollectionState is “ReadyForCollection”. Once it is collected, the state will be updated to “Collected”.

Finally, Algorithm 6 shows the final stage where the pharmacy will execute the PaymentRequest function and that requires the MedicationCollectionState to be “Collected”. Then, the PaymentState is shown as “Pending” and an event is emitted to show the payment request details. Once the PaymentState is “Pending”, the ClaimPayment function is executed by the insurance company, and the paid amount must be equal to the drug’s total cost. If these conditions are met, then the PaymentState will be updated to “Paid” and an event will be broadcast to declare that the claim is paid.

Algorithm 4: Insurance Approval

1. InsuranceApprovalState is a customized enumerate variable that describes different states of the insurance approval request

2. if (caller ∈ RegisteredPharmacies) then
   3. InsuranceApprovalState = pending
   4. Emit an event declaring that Approval is requested
   5. else
   6. Reject.
   7. end

   // Approval is requested by the selected pharmacy
   8. if (caller == Insurance_Company) ∧ (InsuranceApprovalState == pending) then
   9. if ApprovalRequestState == Approved then
   10. Emit an event declaring that prescription has been approved
   11. end
   12. if ApprovalRequestState == Rejected then
   13. Emit an event declaring that prescription has been rejected
   14. end
   15. else
   16. Reject.
   17. end

Algorithm 5: Medication Collection

**Input:** patientID, IPFSHash

1. MedicineCollectionState is a customized enumarate variable that describes different states of the Medicine collection

2. if (caller ∈ RegisteredPharmacies) ∧ (InsuranceApprovalState == Approved) then
   3. MedicationCollectionState = ReadyForDelivery
   4. Emit an event declaring that medication is prepared and ready for delivery
   5. else
   6. Reject.
   7. end

   8. if (caller == patient) ∧ (MedicationCollectionState == ReadyForCollection) then
   9. UpdateMedicationCollectionState = collected
   10. Emit an event declaring medication collection details
   11. else
   12. Reject.
   13. end

   // The patient collects the medicines

VI. TESTING AND VALIDATION

In this section, the main functions of the two smart contracts are tested and validated using the Remix IDE. It is a web-based development environment for
Algorithm 6: Claim Payment

**Input:** MedicationInvoiceID, Drugtotalcost

1. PaymentState is a customized enumarate variable that describes different states of the claim payment.
2. if (caller ∈ RegisteredPharmacies) ∧ (MedicationCollectionState == collected) then
   1. PaymentState = Pending
   2. Emit an event declaring the payment request details
3. else
   4. Reject.
   5. end

// A claim is send to insurance company
6. if (caller == Insurance_Company) ∧ (PaymentState == Pending) ∧ (msg.value == DrugTotalCost) then
   7. Update PaymentState = paid
   8. Emit an event declaring that the claim is paid
9. else
   10. Reject.
   11. end
12. // Insurance company pays the admissible costs

| Table 1. The Ethereum addresses of participants in testing scenario. |
|---------------------------------------------------------------|
| Ethereum Address                                           |
| Regulatory Authority                                       |
| 0x3839DAe701c584544Ceb303FcbB75550d6D4C                  |
| Physician                                                |
| 0x14723A09ACf5D25A0Dcd7a4A4f18D61c600C                   |
| Patient                                                  |
| 0x175F6A6DEBF92277575C9205069C1D6DF8434A372               |
| Pharmacy 1                                               |
| 0x8b051310c5150775a6D7929C4e5364D2DB                     |
| Pharmacy 2                                               |
| 0x167E86A5C7C470F2B99A30337282C267392148                 |
| Pharmacy 3                                               |
| 0x30C6F1E2D78EB8C9a4FAB346F9706773971D9F                |
| Insurance Company                                        |
| 0xa69843F6469C61EEdF98b449A677d33158352b2               |
| RegistrationSCAddress                                    |
| 0x9145CCE82D338254917e481eB46e9943F39138                |

devloping and executing smart contracts. The smart contract code is made publicly available at GitHub. The functionality of smart contracts is evaluated by deploying the registration smart contract and the approval smart contract at “0x9145CC...43F39138” and “0 × 9D7f74d...9e3b5E99”, respectively. The Ethereum addresses of actors that are used in testing are presented in Table 1. The transactions and logs of the main functions are shown below.

- **Insurance Company Registration:** This function can be executed only by the regulatory authority where Ethereum addresses of authorized insurance companies are registered. Figure 5 depicts a successful execution of the insurance company registration function, as well as the corresponding events and logs.

- **Prescription Creation:** This function can be executed by an authorized physician. The physician runs the function by entering several inputs such as patient ID, drug codes, and IPFS hash. The inputs do not reflect a real-life case scenario; they are assumptions used for testing purposes. Figure 6 displays a successful execution of the function and its related events and logs.

- **Insurance Approval:** Once pharmacy 2 sends confirmation, it requests approval from the insurance company to process the prescription by executing RequestInsuranceApproval. The insurance company...
then will either approve or reject the request by running the insurance approval function. Figure 7 shows a successful execution of the insurance approval function.

- **Payment Request**: This function is executed by the selected pharmacy after the medication is collected by the patient. Figure 8 shows a successful payment request.

- **Claim Payment**: This function is executed by the insurance company to pay the admissible costs that are requested by the pharmacy in exchange for the medication given to the patient. The tested result is shown in Figure 9.

VII. DISCUSSION
In this section, we will discuss the security analysis, comparison with existing models, and the generalization of our proposed solution.

A. SECURITY ANALYSIS
In this section, we address major security and privacy concerns related to trust, integrity, availability, and vulnerability to cyber-attacks and how key blockchain properties help to eliminate these issues.

- **Confidentiality and Anonymity**: Confidentiality preserves the privacy of medical data of end-users, where it is disclosed only to authorized entities. In our solution, we employ both on-chain and off-chain confidentiality. On-chain confidentiality is ensured by the deployed private blockchain, where data is shared only with authorized entities using TLS (Transport Layer Security). A trusted certificate authority issues certificates which are utilized between the nodes to establish a secure connection. As for off-chain confidentiality, sensitive data is encrypted using a combination of symmetric and asymmetric cryptography, this is done to preserve features of authenticity and non-repudiation by asymmetric cryptography and speed of symmetric cryptography. The sender would encrypt the data with a symmetric key, and then proceeds to encrypt the symmetric key with the public key of the receiver, such that only the receiver can decrypt it using their private key. Furthermore, identities...
of patients are kept confidential as only the assigned stakeholders are able to view their information. The patients operate under pseudonyms that act as their digital identity with no linkability to their real identity. This does not impede on the transparency and accountability in the system, as timestamped transactions are stored on the blockchain acting as an audit trail which would ensure non-repudiation to the associated pseudonym (EOA) signing the transaction.

- **Integrity and Authenticity**: By leveraging the blockchain in our solution we preserve the integrity through the immutability of blocks, where any changes are easily detected and invalidate the blocks. As such, changes are committed through new sequential transactions, creating a timeline of the changes to the data. Furthermore, each participating node has a keypair which is used to sign transactions. Each participant signs a transaction with their private key which can be decrypted by the public key to ensure authenticity of the transaction and its data. Therefore, non-repudiation of transactions is ensured, and entities will be accountable for the provided data. This is imperative for this application, especially stakeholders such as pharmacies, hospitals, and insurance companies since the processed health insurance claim would be immutable, transparent and auditable, ensuring valid prescription drugs will not be altered by any participant.

- **Availability**: Since the blockchain is a decentralized network with a shared distributed ledger, the services of the system are available as long as there are participants. Data is stored at several nodes making it difficult to disrupt services, where the attacker is required to compromise all the nodes to do so. Furthermore, the system is available for both access of transaction data, as well as generating and submitting transactions to the ledger. As such, the system can run with no-downtime which is detrimental in a critical industry such as healthcare. This ensures that critical cases can be handled and processed without hindrance.

- **Vulnerability to Cyber-Attacks**: Cyber-attacks like Man-In-The-Middle (MITM) and distributed denial-of-service (DDoS) are mitigated through the redundancy of a centralized network [18]. Commonly in centralized systems, compromising the central server denies service of the system. However, in decentralized networks the attacker would have to either flood the network or overload most of the participants, effectively disrupting legitimate service of the system [21]. On the network level, flooding the system effectively would be difficult because of the costs associated with each transaction. Furthermore, the entities are registered on the Registration SC, as such only valid participants have the ability to execute transactions on the network. As such, it is difficult to coordinate a large-scale DDoS attack between valid entities, with non-repudiation ensured with their associated EOA. Therefore, the attack is mitigated through prevention, and the effectiveness of it would be limited to congesting the network rather than denying service, which can also be resolved by removing misbehaving participants.

### B. SECURITY ANALYSIS OF THE SMART CONTRACTS CODES

We perform the security analysis of the smart contracts developed. The Remix IDE provides code-debugging as well as run-time error warnings. However, it is not enough to secure the smart contract. Therefore, to improve the reliability of the smart contracts, a specialized tool named Oyente can be used to test and validate the developed Ethereum smart contracts against vulnerabilities. It establishes trust in the strength of the deployed smart contract by making the code risk-free. Oyente is a Linux-based tool that thoroughly examines the Solidity code to see if it is vulnerable to exploitation and cyber assaults.

Figure 10 shows Oyente’s security results for the registration smart contract, whereas Figure 11 shows the security results for the approval smart contract. The smart contracts are written in Solidity, a high-level programming language. It’s converted into EVM code, which is a low-level stack-based bytecode language [17]. The smart contract’s EVM code coverage was found to be 99.9% and 82.9% respectively, in the security study. The number of opcodes executed divided by the total number of opcodes is the Ethereum Virtual Code coverage percentage. For all types of probable vulnerabilities in the generated smart contract code, the Oyente tool returned “False”.

### C. COMPARISON WITH THE EXISTING SOLUTIONS

To better understand our contribution compared to the state-of-the-art, a comparison shown in Table 2 was made. This comparison was made with respect to critical parameters such as decentralization, security, and including pharmacies as...
participants. We provide a decentralized system that provides resilience and security against the single point of failure problem, which is critical to a healthcare system. A centralized model is more vulnerable to attacks and downtime. Other solutions that provide centralized solutions can’t guarantee continuous run time. Furthermore, observing Table 2 and solutions that provide centralized solutions can’t guarantee resilience and security against the single point of failure problem, hence the system will require off-chain storage. Also, minor modifications are required to be made to the algorithms to adapt to different claims.

Furthermore, our proposed solution can also be used to address other claim abnormalities such as abuse and error due to the intrinsic characteristics of blockchain technology. For instance, unnecessary costs incurred due to the use of inappropriate medical services can be easily detected and traced as all activities and transactions will be approved and stored in the distributed ledger, where all authorized stakeholders in the private network will be able to know exactly who was responsible and when it was carried out due to the immutability and transparency of the data. The same logic applies for claiming abnormalities arising due to errors.

| Features                | [15] | [16] | [7] | [14] | Our Solution |
|-------------------------|------|------|-----|------|--------------|
| Blockchain-Base         | No   | No   | Yes | Yes  | Yes          |
| Smart Contract          | NA   | NA   | No  | Yes  | Yes          |
| Decentralized Storage   | No   | No   | Yes | No   | No           |
| Stability               | No   | No   | Yes | Yes  | Yes          |
| Inclusion Pharmacies    | No   | No   | Yes | No   | Yes          |

VIII. CONCLUSION

In this paper, we proposed processing health insurance claims for prescription drugs using blockchain in a confidential, private, secure, trustworthy, and decentralized manner. The proposed system utilizes a private Ethereum blockchain, where two smart contracts were developed to contribute to recording and logging events automatically. Also, to deal with the limitation of storage, the network was integrated with off-chain storage (IPFS) to store the large-sized data that is expensive to be stored on the blockchain network. A security analysis was conducted to demonstrate the robustness and security of our proposed solution against major security vulnerabilities. The proposed decentralized system provides resilience and security against the single point of failure problem, which is critical to a healthcare system. Lastly, the proposed system can be generalized to serve different health insurance claims systems, including inpatient or outpatient.

In future work, although the proposed blockchain-based approach has considered important issues, there are still some limitations that should be taken into consideration. For instance, the immutability feature in our proposed solution could be an issue because any human error will be permanently stored and cannot be modified or deleted. In addition, the privacy and confidentiality can be improved by using the Quorum platform that supports transaction and smart contract privacy, where it allows creating private transactions that are visible only to specific participants, which is not the case in our approach, as the transaction can be viewed by all authorized entities. Also, it can enhance the performance of the system due to the simplistic consensus algorithms. Furthermore, our solution is planned to be deployed and tested on the real Ethereum network, and an end-to-end DApp will be built to be used by different stakeholders.

REFERENCES

[1] Policybazaar. Health Insurance Claim: Medical Insurance Claim Settlement Ratio. Accessed: Mar. 8, 2022. [Online]. Available: https://www.policybazaar.com/health-insurance-claim/
References

[1] Definitive Healthcare. *Health Insurance Claim 101: What You Need to Know*. Accessed: Mar. 8, 2022. [Online]. Available: https://www.definitivehc.com/blog/medical-claims-101-what-you-need-to-know

[2] R. Zhang, R. Xue, and L. Liu, “Security and privacy on blockchain,” *E. Daniel and F. Tschorsch, “IPFS and friends: A qualitative comparison of blockchain technologies for health insurance claim processing in hospitals,” in Proc. IEEE Conf. Big Data Anal. (ICBDA), Nov. 2017, pp. 99–104.

[3] L. Alhasan, M. Qatawneh, and W. Almobaideen, “Blockchain technology for preventing counterfeit in health insurance,” in *Proc. Int. Conf. Inf. Technol. (ICIT)*, Jul. 2021, pp. 935–941.

[4] I. Alnuaimi et al.: Blockchain-Based Processing of Health Insurance Claims for Prescription Drugs

[5] Khalida Salah (Senior Member, IEEE) received the B.S. degree in computer engineering with a minor in computer science from Iowa State University, USA, in 1990, the M.S. degree in computer systems engineering and the Ph.D. degree in computer science from the Illinois Institute of Technology, USA, in 1994 and 2000, respectively. He is currently a Full Professor at the Department of Electrical and Computer Engineering, Khalifa University, UAE. He joined Khalifa University in August 2010, where he is teaching graduate and undergraduate courses in the areas of cloud computing, computer and network security, computer networks, operating systems, and performance modeling and analysis. Prior to joining Khalifa University, he worked for ten years at the Department of Information and Computer Science, King Fahd University of Petroleum and Minerals (KFUPM), KSA. He has over 190 publications and three patents. He has been giving a number of international keynote speeches, invited talks, tutorials, and research seminars on the subjects of blockchain, the IoT, fog and cloud computing, and cybersecurity. He is a member of IEEE Blockchain Education Committee. He was a recipient of the Khalifa University Outstanding Research Award 2014/2015, the KFUPM University Excellence in Research Award of 2008/09, and the KFUPM Best Research Project Award of 2009/10; and also a recipient of the departmental awards for Distinguished Research and Teaching in prior years. He was the Track Chair of IEEE Globecom 2018 on Cloud Computing. He serves on the Editorial Boards for many WOS-listed journals including *IET Communications*, *IET Networks*, Elsevier’s *JCN*, Wiley’s *JCN*, Wiley’s *JNMM*, *JUCS*, and *AJSE*. He is an Associate Editor of IEEE *Blockchain Newsletter*.

[6] Rajaa Jayaraman received the bachelor’s and master’s degrees in mathematics in India, the Master of Science degree in industrial engineering from New Mexico State University, and the Ph.D. degree in industrial engineering from Texas Tech University, USA. He is an Associate Professor at the Department of Industrial and Systems Engineering, Khalifa University, Abu Dhabi, UAE. His postdoctoral research was on technology adoption and implementation of innovative practices in the healthcare supply chains and service delivery. He has led several successful research projects and pilot implementations of supply chain data standards in the U.S. healthcare systems. His research interests include application of blockchain technology, NFTs, the IoT, process optimization techniques to characterize, model, and study complex systems with applications to supply chains, maintenance planning, and healthcare delivery.

[7] Ilham A. Omar received the bachelor’s degree in electrical and electronic engineering and the master’s degree in engineering systems and management from Khalifa University, United Arab Emirates. She is currently a Research Associate at the Department of Industrial and Systems Engineering, Khalifa University. Her research interests include applications of blockchain, smart contracts, and the IoT technology across various industries.

[8] Ammar Batah received the B.Sc. and M.Sc. degrees in computer engineering from Khalifa University, United Arab Emirates, in 2019 and 2022, respectively. He is currently a Research Associate at the Department of Industrial and Systems Engineering, Khalifa University. His current research interests include blockchain technologies, the Internet of Things (IoT) security, and information security.