IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 11, 2020, accepted April 3, 2020, date of publication April 7, 2020, date of current version April 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2986257

Blockchain-Based Agri-Food Supply Chain: A Complete Solution

AFFAF SHAHID[1], AHMAD ALMOGREN[2], (Senior Member, IEEE), NADEEM JAVAID[1], (Senior Member, IEEE), FAHAD AHMAD AL-ZAHRANI[3], MANSOUR ZUAIR[4], AND MASOOM ALAM[1]

[1]Department of Computer Science, COMSATS University Islamabad, Islamabad 44000, Pakistan
[2]Chair of Cyber Security, Computer Science Department, College of Computer and Information Sciences, King Saud University, Riyadh 11633, Saudi Arabia
[3]Computer Engineering Department, Umm AlQura University, Mecca 24381, Saudi Arabia
[4]Chair of Cyber Security, Computer Engineering Department, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

Corresponding authors: Ahmad Almogren (ahalmogren@ksu.edu.sa) and Nadeem Javaid (nadeemjavaidqau@gmail.com)

This work was supported by the Deanship of Scientific Research, King Saud University, through the Vice Deanship of Scientific Research Chairs.

ABSTRACT Supply chains are evolving into automated and highly complex networks and are becoming an important source of potential benefits in the modern world. At the same time, consumers are now more interested in food product quality. However, it is challenging to track the provenance of data and maintain its traceability throughout the supply chain network. The traditional supply chains are centralized and they depend on a third party for trading. These centralized systems lack transparency, accountability and auditability. In our proposed solution, we have presented a complete solution for blockchain-based Agriculture and Food (Agri-Food) supply chain. It leverages the key features of blockchain and smart contracts, deployed over ethereum blockchain network. Although blockchain provides immutability of data and records in the network, it still fails to solve some major problems in supply chain management like credibility of the involved entities, accountability of the trading process and traceability of the products. Therefore, there is a need of a reliable system that ensures traceability, trust and delivery mechanism in Agri-Food supply chain. In the proposed system, all transactions are written to blockchain which ultimately uploads the data to Interplanetary File Storage System (IPFS). The storage system returns a hash of the data which is stored on blockchain and ensures efficient, secure and reliable solution. Our system provides smart contracts along with their algorithms to show interaction of entities in the system. Furthermore, simulations and evaluation of smart contracts along with the security and vulnerability analyses are also presented in this work.

INDEX TERMS Accountability, blockchain, credibility, reputation, supply chain, traceability, trust.

I. INTRODUCTION

The Supply Chain Management (SCM) is a group of processes and sub-processes carried out for transforming raw material into a final product, maximizing customer value and achieving a maintainable competitive advantage [1]. It is also interpreted as a network of entities that are part of the system from production to trading. The whole supply chain network is divided into several stages. Processes involved in these stages often take months to complete [2]. In such situation, if the final product lacks in quality, it becomes extremely difficult to track the root cause of the problem. The demand for top quality products and interest of end consumers in the provenance of data is increasing rapidly. Therefore, it has become necessary for every supply chain system to track the movement of products from origin to the end consumers [3].

To gain end consumers' trust, the supply chain authorities have to be efficient and accurate in delivering information. It is also important for supply chain authorities to comply with quality, integrity and credibility of the entire supply chain process. Several regulatory authorities have enforced standards for improving quality, transparency and security for supply chain traceability systems. These standards are strictly enforced by the governments of several countries. Canadian government has enforced the use of tags [4] and

The associate editor coordinating the review of this manuscript and approving it for publication was Zhiyi Li.

bar codes to identify the provenance of products. Similar enforcement is also imposed by the Chinese government [1]. The aim of these regulations is to improve transparency of the traceability systems and to ensure high quality of products.

In addition to the responsibility of maintaining traceability, supply chain systems also act as a gateway for trading products. These systems process huge amount of transactional data and thus add more complexity to the network architecture. These networks are generally centralized, so, there is a risk involved for false or inaccurate representation of information [5]. Supply chains allowing financial transactions on their networks, lack trust and credibility due to their centralized service architecture. Moreover, the centralized storage schemes used in supply chain networks are often unable to handle large amount of data leading to unavoidable bottlenecks and therefore, affect the overall performance of the network.

Distributed systems offer fault tolerance, scalability, concurrent processing and better storage schemes. The emergence of blockchain as a base technology of Bitcoin is recognized by several industries around the globe, e.g., finance, Electronic Medical Records (EMRs), Internet of Things (IoT), energy and many more. It is a secure system that overcomes aforementioned risks of centralized systems [6]. However, the current blockchain network is not a one-size-fit all solution, especially for data-driven domains as it faces latency, storage and throughput issues [7]. Several network architectures and distributed consensus protocols that keep the integrity of a blockchain while enabling high throughput and improved storage capabilities, have been presented in [7], [8]. In terms of Agri-Food supply chain, an efficient monitoring of Agri-Food products is critical because of product safety. The growing concerns of consumers and government regarding food quality have also renewed the concept of traceability in supply-chain. However, blockchain plays a significant role in evolution of supply chain with its inherent properties like decentralization, transparency and immutability. Moreover, it also provides smart contracts leveraging safe trading transactions among entities. Despite of the trust less nature of blockchain-based Agri-Food supply chains, it is hard for the end-consumers to trust the product owner and quality of the product before performing a transaction.

Additionally, the traditional centralized storage schemes are unable to handle large amount of data produced during supply chain processes and consequently cause bottleneck. Therefore, several decentralized storage schemes are proposed in literature to overcome the issues like high latency, low throughput and bottlenecks. In [9], a blockchain-based soybean traceability scheme is proposed. The ethereum smart contracts and Interplanetary File Storage System (IPFS) are used to achieve complete traceability in proposed system. IPFS is a popular, decentralized, peer-to-peer file storage system. It uses the technologies like an incentivized block exchange and Distributed Hash Table (DHT). Here, nodes do not trust each other and there is no single point of failure. However, the data stored in IPFS can be accessed easily if its hash available. IPFS nodes also act selfishly while backing up data. Additionally, authors in [9] do not consider the accountability and auditability of trading and delivery of data. Moreover, as Agri-Food supply chains are moving towards e-agriculture; therefore, there's a need of decentralized automated payment mechanism which ensures that the entities in the entire system adhere to the commitment during transaction. Authors in [10] proposed an efficient storage scheme for Agri-Food tracking. The transaction hash in the proposed solution is stored in a secondary database. To retrieve data from IPFS, the transaction hash is accessed from secondary database. Using that transaction hash, IPFS hash is retrieved from the blockchain. However, if the secondary database fails, whole system will fail. Similarly, authors in [11] proposed an auditable protocol for transparent and tamper proof transactions between trading entities. The trading entities are merchants, logistics company and consumers. However, authors have not considered credibility of merchants and trust between trading entities. Additionally, the existing trading networks have information asymmetry between buyers and sellers. The information asymmetry results in poor credibility of trading entities and the end consumers become vulnerable to fraudulent transactions. From the aforementioned limitations of existing works [9]–[11], the following research questions arise:

- Is there a solution available, other than secondary storage, to store transactional hash on IPFS securely while restricting its access to authorized users and ensuring its availability.
- Is it possible that all network nodes have information symmetry while maintaining trust and credibility of nodes.
- Is there a mechanism of decentralized automated payment available which ensure non-repudiation and protect the sellers.
- In a decentralized trading environment, how disputes between two parties will be resolved without any biasness.

Our paper aims to contribute in the growing work on blockchain-based Agri-Food supply chains and provides an end to end solution. To the best of our knowledge, there is no existing work in literature that provides an end to end solution for Agri-food supply chain. However, this paper is an extension of blockchain-based reputation system proposed in [12]. The proposed solution uses ethereum smart contracts to assure an efficient, secure and trusted environment for the supply chain activities. The main contributions of the proposed system are as follows.

- It serves as an end-to-end distributed supply chain system that introduces 1) a traceability scheme and trading and delivery mechanisms, 2) a reputation system for credibility assurance of the entities and 3) an autonomous transaction system.
- It achieves the desired properties of accountability, credibility, auditability, autonomy and authenticity.

- It acts as a better alternative to the existing Agri-Food supply chain systems enabling a scalable and auditable system. It presents the algorithms of smart contracts and evaluates them for vulnerability and their costs of gas and Ethers over the ethereum network.
- It is resilient against well-known attacks and provides inherent important security features. We perform a detailed vulnerability assessment of our system and discuss how our system is robust and ensures security against malicious attacks.

The rest of the paper is organized as follows. Related work and system model are presented in Sections II and III, respectively. Simulations and results are provided in Section IV. Security and vulnerability analyses are discussed in Section V. In Section VI, paper is concluded along with its limitations and future work.

II. RELATED WORK

In this section, we discuss and analyze related schemes proposed for improvement of Agri-Food supply chain system and accentuate the differences to our proposed solution, as sketched in Table 2.

Food safety in recent times is a growing concern for commercial and academic industries. Most of the solutions till date are centralized and result in serious problems such as fraud, tampering and man-in-the-middle attack [13]. Therefore, literature has introduced several blockchain-based traceability and information security in Agri-Food supply chain systems. Hereof, author in [14] has proposed a traceability scheme based on Hazard Analysis and Critical Control Points (HACCP), blockchain and IoT. Furthermore, blockchain along with its advantages has some disadvantages as well, i.e., it lacks scalability when data increases to a certain level. In this regard, BigChainDB is used to fill the gap which provides a scalable solution. The proposed solution is then applied to an example scenario to show the significant transparency and efficiency and how it favours HACCP regulations. However, the proposed scheme does not specify the current ownership details of products. In addition to this, a case study on product traceability is presented in [15]. According to the authors, tracing the provenance of products in supply chain must be transparent, tamper-proof and adaptive to the changing environments. Therefore, they have designed an origin-chain that uses private and public blockchain. As blockchain has limited storage, origin-chain stores the data on-chain and off-chain. On-chain storage includes the hashes of data while off-chain storage has the raw files and addresses of smart contracts. The authors have also provided a case study with actual implementation and deployment of origin-chain in industry. Additionally, they have also discussed the adaptability of the solution and concluded that the blockchain is a good option for traceability in SCM. However, security and privacy are the main concerns. In this regard, authors in [16] have introduced blockchain-based food information security in SCM. According to them, a no. of solutions have been provided to achieve traceability. However, these solutions are not able to achieve accurate traceability required for Chinese market.Based on the hypothetical conclusions and analysis, authors have provided a more reliable and efficient solutions. However, the practical implementation of the whole solution still lacks behind. In [17], authors have proposed blockchain-based decentralized traceability process and provided a case study. However, auditability and integrity is compromised. Considering the food safety issues, blockchain and IoT based solution is proposed for Agri-Food supply chain and information security [18]. They created a use case for traceability of product from farm to the table and compared the results using different implementation platforms, i.e. ethereum and hyperledger. Authors in [19] reviewed the concepts of information and communication technology and blockchain. They proposed an e-agriculture system and evaluation tool. This system can be used to get the certain requirements for blockchain-based agriculture systems. However, the proposed system lacks in terms of practical implementation and feasibility of applying in real-environment. Authors have proposed a blockchain-based anonymity preserving delivery mechanism for physical items in [20]. They have achieved anonymity, fairness and unlinkability of buyers and sellers. However, authors compromised the accountability of the entities involved. Authors in [21] have tackled the problem of some how cloning the RFID tags post delivery. In this regards, they used Bitcoin's blockchain and implemented proof-of-concept. After the performance evaluation of the system, authors concluded that the cost of managing product ownership is reduced to 1 USD for almost six transfers.

In traditional storage schemes, the data is stored in centralized storage. After the invention of blockchain, many decentralized storage systems are used to store the data in a decentralized manner. In [10], authors proposed an efficient storage scheme for Agri-Food product tacking. Authors used IPFS along with secondary database to achieve the traceability. IPFS is a network used to store and share data in a decentralized file system. To retrieve data from IPFS, the transaction hash is accessed from secondary database. Using that transaction hash, IPFS hash is retrieved from the blockchain. However, if the secondary database fails, whole system will fail. Authors in [9] have proposed an approach for efficient transactions of soybean traceability in Agri-Food supply chain. The proposed solution overcomes the problems of centralized solutions and eliminates the need for a trusted third party. It maintains high integrity, reliability and more security. However, authors have not considered the accountability and auditability of the data delivered and automated payments. Authors have proposed a proof of delivery mechanism to deliver physical assets between multiple transporters in [22]. In the proposed solution, all the entities act honestly by incentivizing the trading entities. Automated payments through ethers are also part of the proposed solution. However, the proposed scheme has used a key and asset while transporting the asset. The key and the asset delivered have no relation between them and as a result, transporters can easily tamper the asset to be delivered. Paper [11] has proposed an auditable protocol for transparent, tamper-proof and verifiable transactions between entities. It has also proposed a pre-verification technique to overcome the limitation in paper [22] However, credibility of merchants is not considered. In [23], a decentralized storage mechanism along with ethereum blockchain is proposed. The paper aims at overcoming the risks of centralized storage, i.e., leakage of the sensitive data and a single point of failure. The decentralized storage mechanism used is IPFS. In the proposed framework, before storing the data in IPFS, the file is encrypted using a file encryption algorithm. The cipher text obtained after encryption is uploaded to IPFS. IPFS provides the hash of the stored file which is recorded in ethereum blockchain. However, the proposed solution if applied in Internet of Things (IoT) scenario, will not work efficiently because of the increased computational overhead. The authors in [5] have proposed an approach for trustless privacy preserving reputation system. The proposed solution maintains the anonymous ratings of products and provides the correctness and security analysis of the proposed scheme. However, no performance analysis is provided that is required to guarantee efficiency of token generation. Additionally, there is no link between the ratings and the transactions and are consequently prone to malicious users. In [24], authors have discussed the existing and proposed solutions that work on maintaining trust and reputation while performing online transactions. The authors have then proposed an agenda for reputation systems. Every online transactional platform has the problem of truthfulness between the seller and buyer. This is for the reason that the both parties do not directly meet to perform the transaction. Therefore, a review system is required that maintains the sellers reviews against his profile and helps buyers to evaluate the seller and products beforehand. A blockchain-based reputation system is proposed in [25]. The proposed solution provides data credibility in the vehicular networks. Each message that is transfered from one entity to another is reviewed on the basis of environment traffic. Each review is then blocked and chained in the network. Based on these reviews, the vehicles are able to trust the messages obtained. However, the credibility of the entities that transfer messages is not maintained. Concluding from the above literature, use of blockchain in agriculture based supply chain system is growing exponentially. It is adopted to enhance the transparency, traceability and food safety issues in the present supply chain systems. Therefore, by getting the motivation from the above literature, we have proposed a blockchain-based solution for maintaining accountability, auditability and credibility in Agri-Food supply chain systems.

III. SYSTEM MODEL

In this section, we describe our proposed solution. We have provided a traceability scheme for digitally tracking Agri-Food products from origin to end consumers. Our system introduces a trading and delivery mechanism to allow secure trading between entities of Agri-Food supply chain. A reputation system is also used for the credibility assurance of these entities. The proposed model follows a layered architecture and is categorized into three layers. The first layer, i.e., data layer, handles the interactions between entities of Agri-food supply chains. These interactions involve the trading of products along with a proof of an auditable delivery. The second layer is the blockchain layer that handles the transactional data of the trading and delivery events. Also, it keeps track of the reputation of the entities involved in the system. To improve storage capabilities, the blockchain layer only keeps the hashes of the data and the actual data is stored on the third layer, i.e., storage layer. The blockchain layer enforces strict access control strategies to prevent unauthorized reads and writes to the storage layer. The third layer is essentially the storage layer and is solely responsible for storing the transactions' and events' data of blockchain on IPFS. As, IPFS is a decentralized storage medium, it leverages the proposed system with high throughput, low latency and scalability [26].

The sub-sections below elaborate how the proposed system achieves traceability. They also explain the trade events between the Agri-Food supply chain entities and the delivery mechanism that provides an auditable delivery of products. Lastly, they define how the reputation system works and benefits the proposed system.

A. TRACEABILITY

Supply chain systems involve a large number of entities to carry out the entire process of production and transportation of Agri-Food products from origin to the end consumers. Therefore, it is cumbersome to track and trace the entire process. In order to achieve complete traceability, we record the trading transaction from initiation, add the product's unique identity and lot number to each succeeding transaction and record the hashes to maintain hash chain.[1] Lot is a group of products to be traded in a warehouse and lot number is the unique identifier for these groups of products. For maintaining the hash chain, the transactional data is stored in IPFS. The hashes of data are recorded in ethereum blockchain which overcomes the limitation of IPFS. In order to write or access data from blockchain, access control strategy is applied which ensures the privacy and confidentiality in the network. The access control strategies make sure that the transaction is carried out by the authorized user. Only the registered users are allowed to perform the specific transaction. Moreover, each function in the smart contract is allowed to be executed by specific entities. No unauthorized entities are allowed to perform any task. Multiple supply chain entities are registered in the system that interact through smart contracts, algorithm 1 represents the entity registration process. This process, takes *entityAddress* and *entityType* as input parameters and as a result, registers the respective entity as

[1] Hash chain is the list of IPFS hashes recorded in blockchain.

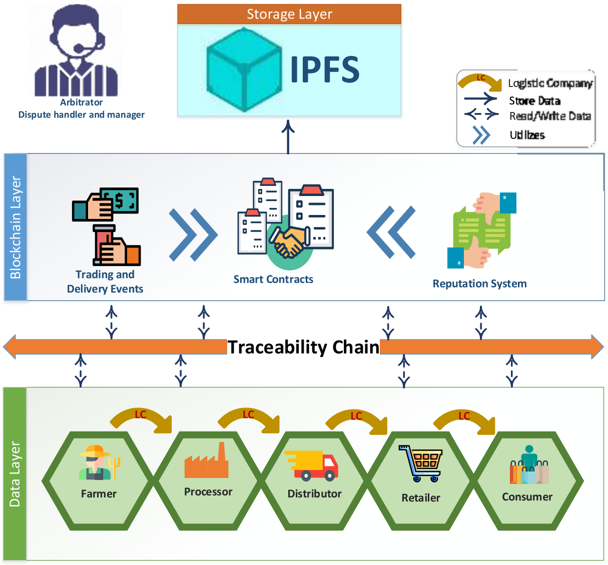


FIGURE 1. Blockchain-based end to end solution for agri-food supply chain.

Algorithm 1 Add Entity

```
procedure AddEntity(...params)
    params[0] ← entityAddress
    params[1] ← entityType
    Arbitrator ← 0 × 49e...F6d ▷ Number of total lots
    struct IDENTITY
        [type: string]: STRING
    end struct
    if message.sender == Arbitrator then
        if keccak256(entityType) == keccak256('Enti-
        ty's initials') then
            Identity[addr] ← 'EntityName'
        end if
    end if
end procedure
```

an authorized user of the system. These entities are part of data layer and are described as follows.

- **Farmer:** A farmer is the first entity in Agri-Food supply chain and is the first one to invoke smart contract for trading. Farmer produces large amount of crops and take the responsibility for assuring and monitoring the crops' growth details. He sells these crops to the processors.
- **Processor:** A processor buys the crops from farmers. He is responsible for eliminating extra material from the crops and converting them into a finalized product. Processor sells this finalized product to distributors.
- **Distributor:** A distributor maintains a warehouse by buying finalized products from processors and is responsible for selling it to the retailers.
- **Retailer:** A retailer buys the finished traceable products from distributors and sells them to customers in smaller quantities. Traceable product refers to specific identifiers of the goods that allow tracking the provenance data.
- **Consumer:** Consumer is an end user who buys and consumes the products from retailers. A consumer verifies the credibility of a seller through reputation system before buying the products.
- **Logistic Company** Logistic Company (LC) is responsible for an auditable delivery of the products from product owners to the purchasers.
- **Arbitrator** Arbitrator is an off-chain entity that is selected to monitor and manage the entire network. Additionally, it also acts as dispute handler.

The traceability process consists of three smart contracts, i.e., Registration Contract (RC), Add to Lot Contract (ALC) and Add Transaction Contract (ATC). These contracts require the addresses of each preceding contracts in order to maintain traceability chain of transactions. For this purpose, all three contracts are deployed in order to get their respective addresses. In this way, the solution helps the end consumers to achieve complete traceability and maintain data provenance. The RC is used to register the Agri-Food supply chain entities and the products available by each entity. The product registration process as shown in algorithm 2, contains the address of ALC to add the lot details of the product. AddLot function is described in algorithm 3. It takes *productLot*, *materialLot* and *addTransAddress* as input. Parameter *AddTransAddress* is the contract address of AddTransaction contract. Similarly, every ALC has the link address to the ATC. Additionally, the supply chain entities are registered to the network and authorization is performed whenever an event takes place. Only the specified entities are allowed to perform certain transactions, for instance, only arbitrators have the authority to remove an entity or malicious users from the system. Arbitrators are the off-chain entities that are responsible for dispute handling.

All transactions of product registration, lot addition and updating transactions' hashes are permanently stored on blockchain. The owner of the product is responsible to deploy ALC along with the RC. ALC contains the details of products, lot information and transaction data. The transactions are written to the blockchain in order to make sure that a product is successfully transferred to the next entity in the Agri-Food supply chain. Moreover, the traceability scheme ensures that all transactions are successfully chained to blockchain. So, in this way complete process of tracing the provenance of data is ensured.

B. TRADING AND DELIVERY

Before discussing the trading and delivery mechanism, lets consider a scenario when end consumer has not yet initialized the transaction and wants to know the market reputation of traders. For this purpose, reputation system is proposed as

Algorithm 2 Registration

```
procedure RegisterProduct(...params)
    params[0] ← productName
    params[1] ← productCode
    params[2] ← addToLotAddress
    n ← 1                                  ▷ Number of total lots
    struct PRODUCT
        $productName: STRING
        $productCode: STRING
        $productOwner: ADDRESS
        $addToLotAddress: ADDRESS
        $addTime: UINT
    end struct
    AUTHORIZED ← Authorized Users
    if message.sender ϵ AUTHORIZED then
        Lot[n][$productName] ← param[0]
        Lot[n][$productCode] ← param[1]
        Lot[n][$productOwner] ← message.sender
        Lot[n][$productLotAddress] ← param[2]
        Lot[n][$addTime] ← Date.now()
        n++
    end if
end procedure
```

Algorithm 3 Add Transaction

```
procedure ADDLOT(...params)
    params[0] ← productLot
    params[1] ← materialLot
    params[2] ← addTransAddress
    n ← 1                                  ▷ Number of total lots
    struct LOT
        $productLot: STRING
        $materialLot: STRING
        $lotAdmin: ADDRESS
        $addTransferAddress: ADDRESS
        $addTime: UINT
    end struct
    AUTHORIZED ← Authorized Users
    if message.sender ϵ AUTHORIZED then
        Lot[n][$productLot] ← param[0]
        Lot[n][$materialLot] ← param[1]
        Lot[n][$lotAdmin] ← message.sender
        Lot[n][$addTransferAddress] ← param[2]
        Lot[n][$addTime] ← Date.now()
        n++
    end if
end procedure
```

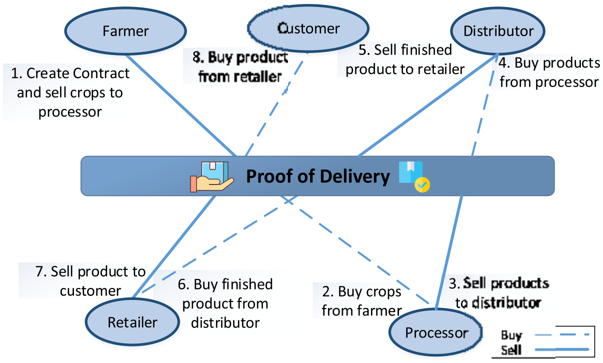

Farmer
Customer
Distributor
8. Buy product from retailer
5. Sell finished product to retailer
4. Buy products from processor
1. Create Contract and sell crops to processor
Proof of Delivery
7. Sell product to customer
6. Buy finished product from distributor
2. Buy crops from farmer
3. Sell products to distributor
Retailer
Processor
Buy
Sell

FIGURE 2. Trading and delivery mechanism.

described in Section III-C. This system ensures that the product owners are reliable enough to trust. Furthermore, for the delivery of product from one entity to another, it is made sure that the whole process can be tracked and traced by recording the information on blockchain. It ensures auditable delivery to end consumer. There are three main entities that are involved in the trading and delivery mechanism, i.e., product owner, LC and purchaser. Where, product owner is the one who sells the product in supply chain; LC is the courier service that transfers the goods; and purchaser, as name depicts, is the one who wants to spend ethers to buy a product. The LC as mentioned earlier is a registered entity of the system. In case of disputes during transactions, arbitrators are responsible for off-chain settlement of the disputes. However, Figure 2 represents the trading and delivery model.

In order to carry out the trading process, at first, the trading entities are registered to the smart contract,i.e., RC and authenticated using their ethereum addresses. After that, the contract between product owner and purchaser is initiated. Here, the purchaser selects the product and enters its code, i.e., P_1, which is used as a unique identifier for that product and used by LC. The product code along with the product details are sent to the product owner. Moreover, the product details, i.e., product owner, picture and price are uploaded to IPFS and in return, IPFS hash is received. This hash helps in proving the authenticity of the product being transferred. After receiving the product, the purchaser confirms the successful delivery and logsitic company gets paid, as described in algorithm 4. Furthermore, in order to confirm the trading transaction, both parties pay security amount to the contract. The security function as shown in algorithm 5, deposits the due amount along with the fine in order to ensure successful completion of the trading and delivery process. The additional amount of fine ensures punishment by arbitrator in case of dispute. Once the transaction is confirmed, the purchaser submits his payment amount to the product owner. In case of dispute, all funds are transferred to the arbitrator's account and distributed according to the off-chain settlement of the dispute.

Furthermore, after the complete transaction between the product owner and purchaser, the smart contract between product owner and LC is initiated. This contract manages the transportation of products from one location to another. This process also collects the transportation security from both parties, i.e. product owner and purchaser. This security amount is collected from both parties to avoid the fear

Algorithm 4 Add to Lot

```
procedure ADDT$_R$(...params)
    params[0] ← currentTranHash
    params[1] ← previousTranHash
    params[2] ← recieverAddress
    n ← 1                                    ▷ Number of transactions
    struct LOT
        $currentTranHash: STRING
        $previousTranHash: STRING
        $senderAddress: ADDRESS
        $recieverAddress: ADDRESS
        $addTime: UINT
    end struct
    AUTHORIZED ← Authorized Users
    if message.sender ϵ AUTHORIZED then
        Lot[n][$currentTranHash] ← param[0]
        Lot[n][$previousTranHash] ← param[1]
        Lot[n][$senderAddress] ← message.sender
        Lot[n][$recieverAddress] ← param[2]
        Lot[n][$addTime] ← Date.now()
        n++
    end if
end procedure
```

Algorithm 5 Product Owner and Purchaser (POandP)

```
procedure DepositSecurity(PO_Addr, P_Addr, d-
    eposit)
    Output: Security deposited sucessfully
    if msg.sender == PO_addr then
        Deposit LCfee + fine
    else
        if msg.sender == P_addr then
            Deposit ProductPrice + LCfee + fine
        else
            return Cancel Transaction
        end if
    end if
end procedure
procedure PO_Payment(string k)
    if keccak256(k) == (hashk1)
        PO_Add.transfer(ProductPrice + fine)
        P_Add.transfer(fine)
    end if
procedure
```

of manipulation. When a LC collects the product for delivery, a pre-verification process is carried out. In this process, the IPFS hash is used to access the product details and match them with the actual product. This process makes sure that the LC does not change the product during the delivery process. Consequently, it ensures the authenticity of the product as well.

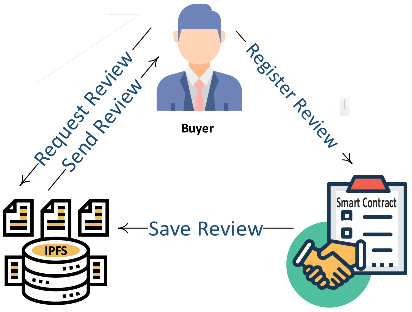


FIGURE 3. Reputation system.

Algorithm 6 Product Owner and LC (POandLC)

```
procedure ProductOwnerAndLogisticCompany
(address)
    Arbitrator ← 0 × 49e...F6d
    ProductIPFS ← Phash
    HASH K1 ← 0 × 0a75b8
    productPrice ← 0.01                      ▷ In Ethers
    LCFee ← 0.01                             ▷ In Ethers
    FINE ← 0.01                              ▷ In Ethers
    TOTAL BALANCE                ▷ Total account balance
    time ← Date.now()                     ▷ Current time
    struct LogisticCompaney
        transfer(balance: double): VOID
    end struct
    struct ProductOwner
        transfer(balance: double): VOID
    end struct
    if mssage.sender == LogosticCompaneyAddress &&
    keccak256(address) == HASHK1 then
        balance ← productPrice + LCFee + FINE
        LogisticCompany.transfer(balance)
        ProductOwner.transfer(TotalBalance)
    end if
end procedure
```

C. REPUTATION SYSTEM

A reputation system, as shown in Figure 3, is introduced in blockchain layer of the proposed model. The reputation system is responsible for assuring the credibility of product owner and the assets delivered. It maintains the immutability and integrity of the reviews registered in the system. In contrast to the traditional reputation systems, the reviews are recorded in IPFS while their hashes are stored in blockchain. In this way, immutability and integrity of reviews are maintained. Reputation contract is triggered after the trade events occur between the buyers and sellers. The proposed system is responsible for invoking smart contracts to provide service based reviews to the sellers. Algorithm 6 describes the reviews registration process. Once the transaction is completed, the buyers register reviews. For the next transaction, a buyer requests reviews of sellers and performs transaction on the basis of those reviews. Moreover, the trading entities in the proposed solution act as both seller and buyer except the farmer and end customer. For instance, the retailer as shown in 2, buys the end product from distributor and sells it to the customers. Similarly, processor buys the crops from farmers and after some processing, sells the end product to the distributor. The terms sellers and product owners, and buyers and purchasers are used alternatively throughout the paper.

The reputation system provides the trust values to the sellers in order to increase the trust among trading entities. Whenever an entity buys a product from product owner, it decides the ratings and provides a reviews for the product owner. The trust values are the quality ratings of the services provided by the sellers. Reputation of an entity either increases or decreases based on the trust values stored in blockchain-based supply-chain. When the trust value of a seller is high, it means that the seller is highly trustworthy. Moreover, based on the trust values of sellers, purchaser decides whether the product owner is reliable or not. However, a seller may have some positive and negative ratings. Therefore, the trust value in the proposed solution is calculated using Equation (1). Where, $\sum Ratings$ denotes sum of all ratings of a seller and $Total_{Rev}$ is the total number of reviews provided to the seller.

$$TrustValue = \sum Ratings / Total_{Rev} \tag{1}$$

The proposed system also ensures trust among the trading entities and make sure that the buyer knows the reputation before purchasing the product from seller. Whenever the entities sign a smart contract for trading, the smart contract for reputation is also triggered that provides the reviews of available data sellers. Once the trading is successfully performed, the buyer also registers review for the seller on the basis of the products received. The review registered by the buyer is then stored against the seller's profile in blockchain system. The smart contract as shown in Figure 3 utilizes four functions, i.e., RegisterReview(), SearchReview(), SaveReview, SendReview() and RequestReview(). These five functions are responsible for registering, checking content and ensuring the existence of a review, saving, sending and requesting for a new review, respectively. The RegisterReview() function takes metadata, ratings of the asset and review details as input. These values are then used by the end consumers to evaluate the quality of the product and reputation of the product owner.

Algorithm 7 Register Review

```
procedure RegisterReview(reviewContent, rating,
msg.sender)
    Output: Review registered successfully
    if msg.sender is an element of Authorization list
    && msg.sender == P_address then
        Add review content and ratings against PO_Addr
        ReviewCounter++
    else
        Revert transaction and display error
    end if
procedure SaveReview()
procedure SendReview()
procedure RequestReview()
procedure SearchReview()
```

IV. SIMULATIONS AND RESULTS

In this section, we discuss the assumption, simulation tools and performance results of the proposed system. Following are some important assumptions:

- the arbitrators of the system are honest entities and they do not take biased decisions while resolving a dispute,
- arbitrators have high computational power than other involved entities of the network,
- no entity on the network has enough computational power to compromise more than half of the network nodes and
- only registered entities can buy or sell products in the market.

For the simulations purpose, an open-source platform for blockchain, i.e., ethereum is used. The smart contracts are deployed over ethereum test network "Rinkeyby" [27]. Ethereum uses blockchain technology to develop decentralized applications. To assess the performance of blockchain-based supply-chain network, we have used Remix Integrated Development Environment (IDE) [28], Ganache [29] and Metamask [30]. Remix facilitates in writing, executing and testing a smart contract. The language used for writing a smart contract in Remix is Solidity. Whereas, Ganache provides the virtual accounts with pre-defined amount of crypto-currency. After each transaction, the crypto-currency is deducted from the account that performs the transaction. Each account in Ganache has its own private key and unique address. However, Metamask is an extension in browser that acts as a bridge between Ganache and Remix IDE and helps to establish connection between them.

The specifications of the system are: intel core i5, 2.4 GHz processor, 8 GB RAM and 500 GB storage. The performance parameters used to evaluate proposed solution's performance are as follows.

- transaction and execution cost (gas) of smart contracts,
- total amount of gas consumed for input strings with different lengths in review system,
- mining time for input strings of same and different lengths in reputation system,
- deployment cost of smart contracts and
- different number of products during registration in Agri-Food supply chain.

In Figure 4, the gas consumption of reputation system smart contract is shown. The reputation system consists of four functions, i.e., RegisterReview(), SearchRatings(), SearchReview() and DoesReviewExist(). It is clearly visible

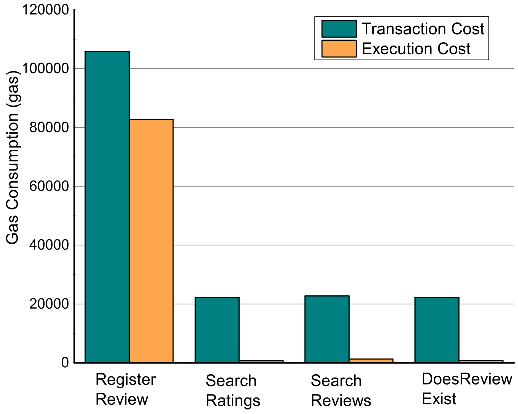


FIGURE 4. Gas consumption of reputation system.

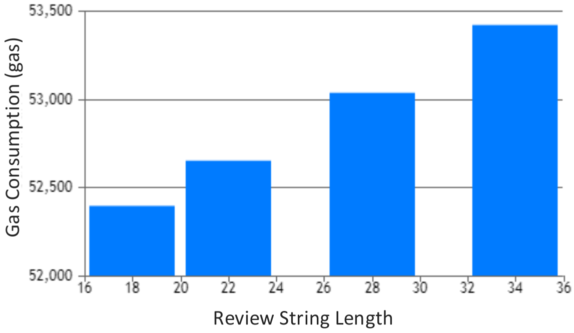


FIGURE 5. Gas consumption against input string length.

from the graph that RegisterReview() function consumes the maximum amount of gas for execution and transaction as compared to the other functions. This is because the RegisterReview() function is responsible for saving the reviews against the user's profile in blockchain and performing logically complex operations. Therefore, the transactional costs for other functions are relatively less. The execution cost depends on computational complexity of the transactions. The transaction cost is the combination of execution cost and the cost of sending smart contract code to the ethereum blockchain.

Figure 5, shows the gas consumption against the input strings of different lengths provided for each review. By plotting the graph of these input strings, we conclude that the relation between gas consumption and length of an input string is directly proportional, i.e., by increasing the length of an input string, the gas consumption of the respective string also increases. Therefore, we can say that longer reviews will cost more as compared to the shorter ones.

In order to compare the mining time of reviews against the input string length, we plotted a bar graph as shown in Figure 6. The input values provided in the reputation system are processed as strings. We provided the input values of different lengths and investigated their effect on mining time for each string. It is observed that the mining time for input is totally different and different lengths of input strings do not effect the mining time. However, the mining time of a transaction is dependent on the complexity of mining process in a network. Miners in a network are responsible for calculating a nonce that must be less than a target value. Hence, if the target value has more difficulty level, mining time will also increase and vice versa.

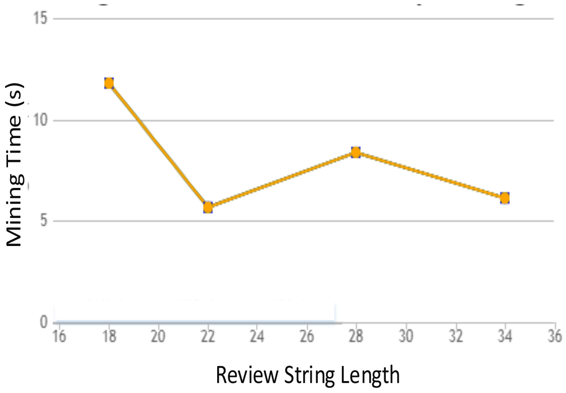


FIGURE 6. Mining time against different input length.

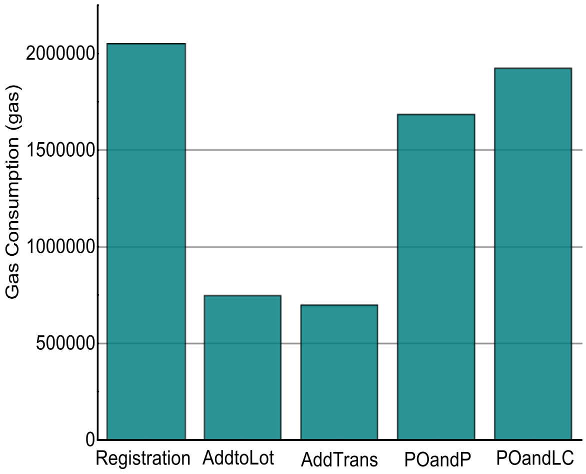


FIGURE 7. Gas consumption of smart contracts.

TABLE 1. Smart contracts cost test (gas price = 1 Gwei, 1 ether = 176.52 USD).

Smart Contract	Deployment Gas	Cost in Ether	USD
Registration	1744760	0.0017448	$0.30708
Add to Lot	748260	0.0007483	$0.1317
Add Transaction	700898	0.0007009	$0.12336
POandP	1683003	0.001683	$0.29621
POandLC	1916209	0.0019162	$0.33725

We evaluate the gas consumption for the deployment of each contract in our proposed solution. The gas consumption is presented in Figure 7. However, actual and USD cost of smart contracts is shown in the Table 1.

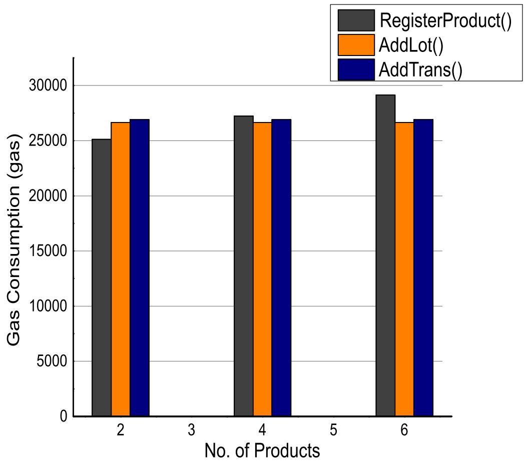


FIGURE 8. No. of products vs gas consumption.

Moreover, we tested different functions of the smart contracts with random values. It is observed from experiments that by increasing the number of products during the registration process, the RegisterProduct() function costs more as compared to the other functions like AddLot() and AddTrans(). The Figure 8 represents the increase in gas consumption of the RegisterProduct() function. However, the gas consumption of other functions is not affected by the increase in number of products.

V. SECURITY ANALYSES

With the growing use of blockchain technology in different industries, various types of attacks have emerged. Therefore, this section provides the vulnerability, robustness and security analyses of the proposed solution and briefs how it is resilient against different attacks.

A. VULNERABILITY AND ROBUSTNESS

A smart contract is one of the main building blocks of the blockchian. The solution proposed in this paper uses smart multiple contracts for transactional activities. Smart contract is a piece of executable code that is responsible to facilitate digital transactions and provide security in terms of authenticity, credibility and immutability. Once a smart contract is deployed on the blockchain then it cannot be modified and executed as it is. If its code is vulnerable, it seriously affects the security of a blockchain [31].

As smart contracts are not mature enough; therefore, they still has some security gaps. During ethereum's initial development, Decentralized Autonomous Organization (DAO) hack was one of the biggest hacking incident which caused the loss of 3.6 million ethers that are equivalent to $70 million. DAO was responsible for decentralized financial transactions via smart contracts [32]. Similarly, many other vulnerabilities in smart contracts like re-entrancy bugs, time dependency, concurrency bugs and callstack attacks are introduced in [32]. These vulnerabilities cause significant losses to blockchain-based systems. The well-known vulnerabilities are defined as follows.

- Call Stack Attack: This attack is also known as call depth attack. It is stated that if a call depth is equal to 1024 frames, the calling function will fail and only run if the depth is 1023 frames. It is also stated that if a call or send function is used to call another contract, the call depth increases by one. So, if a call function calls itself for 1023 times and makes the call stack limit 1024, the next instruction fails.
- Time Dependency Attack: This attack is miner centric attack in which a miner manipulates the conditions of timestamp to favor himself. While mining a transaction, miner set the timestamp of the transaction. In general, a timestamp is time of miner's physical machine.
- Concurrency Bug: It is also a miner side bug and comes under the umbrella of transaction ordering dependency. It arises when two functions are executed at the same time. This problem often comes up when a data structure or database is updated.
- Re-entrancy Vulnerability: This vulnerability refers to a well known DAO attack described earlier. It makes use of path conditions in order to check for re-entrany conditions. In ethereum, when a contract calls another contract, the current transaction waits for the call to finish. This issue can lead to a situation when a transaction makes use of intermediate state of the caller.

Therefore, there is a need to analyze smart contract codes in order to make the system robust against aforementioned attacks. Oyente is an open source security analyzer of ethereum smart contracts proposed by authors in [32]. It analyzes the smart contract on the basis of symbolic execution paths; where, each path has a certain condition. The main responsibilities of oyente are as follows.

- exploring all possible execution paths by using dummy values for variables,
- recording contract's behavior in each path,
- summarizing the conditions for each path and
- checking for violation of any property.

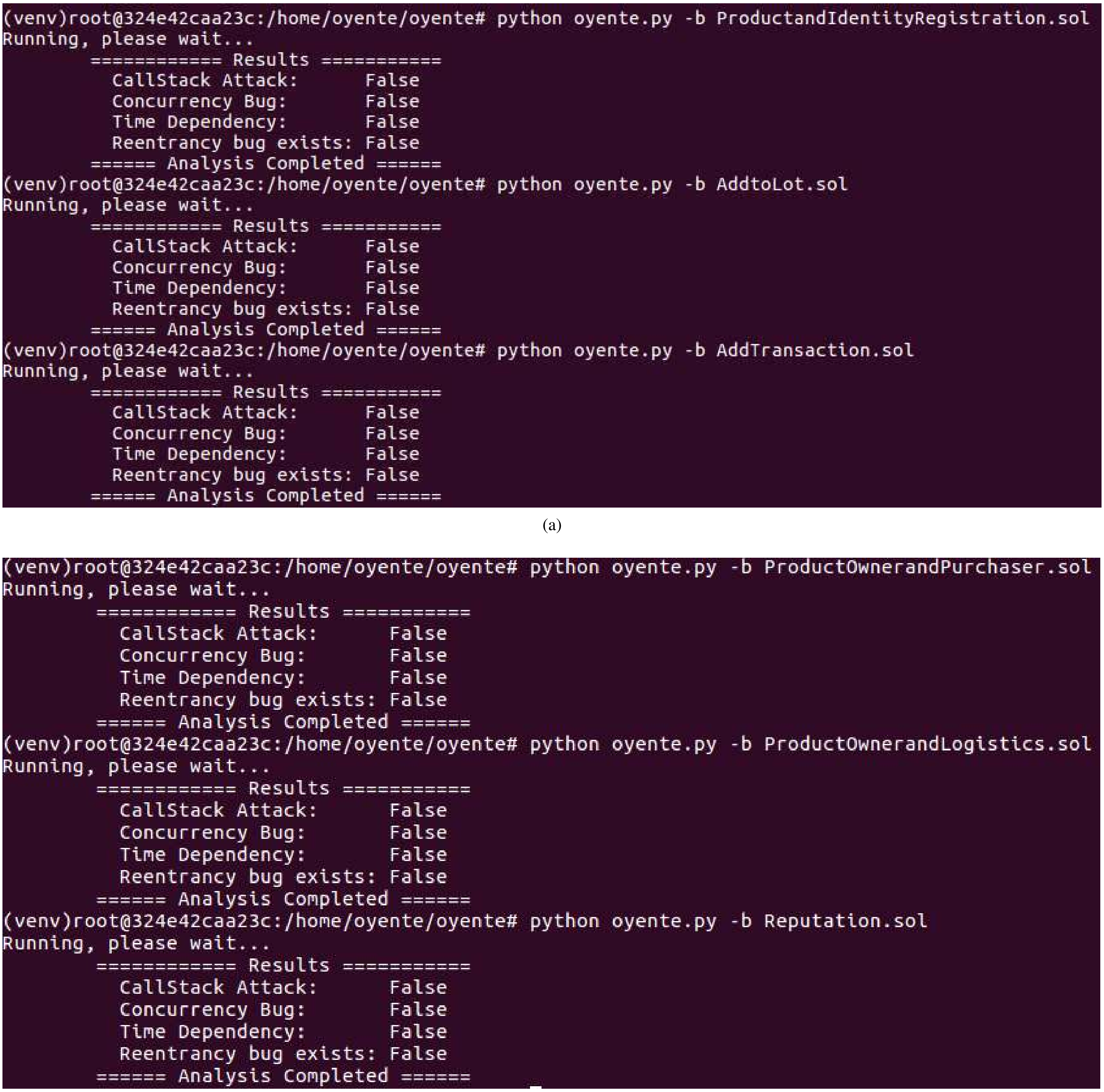
According to an analysis in [32], oyente has flagged 8,833 smart contracts out of 19,366 as vulnerable including the DAO smart contract. We tested our smart contracts with oyente for any security and vulnerability attacks. The Figure 9 clearly shows that all contracts are robust against the aforementioned vulnerability attacks and all reported results are false.

B. SECURITY THEOREMS

This section presents security theorems and features of the proposed solution. It also provides comparison with related schemes as shown in Table 2.

Theorem 1 (Suppose That the Proposed System Does Not Respond or Keeps Denying the Transaction):

Proof: The probability of denial of services is nearly impossible. The proposed system uses smart contracts that



```
(venv)root@324e42caa23c:/home/oyente/oyente# python oyente.py -b ProductandIdentityRegistration.sol
Running, please wait...
        ============ Results ===========
          CallStack Attack:      False
          Concurrency Bug:       False
          Time Dependency:       False
          Reentrancy bug exists: False
        ====== Analysis Completed ======
(venv)root@324e42caa23c:/home/oyente/oyente# python oyente.py -b AddtoLot.sol
Running, please wait...
        ============ Results ===========
          CallStack Attack:      False
          Concurrency Bug:       False
          Time Dependency:       False
          Reentrancy bug exists: False
        ====== Analysis Completed ======
(venv)root@324e42caa23c:/home/oyente/oyente# python oyente.py -b AddTransaction.sol
Running, please wait...
        ============ Results ===========
          CallStack Attack:      False
          Concurrency Bug:       False
          Time Dependency:       False
          Reentrancy bug exists: False
        ====== Analysis Completed ======
```

(a)

```
(venv)root@324e42caa23c:/home/oyente/oyente# python oyente.py -b ProductOwnerandPurchaser.sol
Running, please wait...
        ============ Results ===========
          CallStack Attack:      False
          Concurrency Bug:       False
          Time Dependency:       False
          Reentrancy bug exists: False
        ====== Analysis Completed ======
(venv)root@324e42caa23c:/home/oyente/oyente# python oyente.py -b ProductOwnerandLogistics.sol
Running, please wait...
        ============ Results ===========
          CallStack Attack:      False
          Concurrency Bug:       False
          Time Dependency:       False
          Reentrancy bug exists: False
        ====== Analysis Completed ======
(venv)root@324e42caa23c:/home/oyente/oyente# python oyente.py -b Reputation.sol
Running, please wait...
        ============ Results ===========
          CallStack Attack:      False
          Concurrency Bug:       False
          Time Dependency:       False
          Reentrancy bug exists: False
        ====== Analysis Completed ======
```

(b)

FIGURE 9. Vulnerability analysis of smart contracts.

are deployed over ethereum network. The smart contracts make sure the availability of services to customers. Moreover, ethereum is a distributed ledger that maintains several nodes and follows a consensus mechanism. These mechanisms make the distributed ledgers highly robust against denial of service attacks.

Theorem 2 (Assume That a Dishonest LC Tries to Change the Product While Delivering and Deny Their Action):

Proof: The product hash verification is done when the product arrives at LC. It makes it impossible for the LC to change the product. This pre-verification step ensures that the products are consistent with the customer's orders. Moreover, as blockchain is an immutable ledger, it makes sure that no entity denies his action. Additionally, ethereum uses Elliptic Curve Digital Signature Algorithm (ECDSA) for encryption which signs all transactions. Consequently,

TABLE 2. Comparison of proposed solution with existing techniques.

Features	Soybean Traceability [10]	Efficient Product Tracking [11]	Proof of Delivery [23]	Blockchain-Driven IoT and Consensus Mechanism [34]	Boundary Conditions for Traceability [35]	Our Solution
Traceability	✓	✓	✓	✓	✓	✓
Accountability	✓	✓	✓	✓	✓	✓
Credibility	✗	✗	✗	✓	✗	✓
Authenticity of Product	✗	✗	✗	✗	✗	✓
Automated Payments	✗	✗	✓	✓	✓	✓
Delivery Mechanism	✗	✗	✓	✓	✗	✓

it becomes unlikely for the malicious users to deny their actions.

C. MANAGERIAL IMPLICATIONS

The practical implementation of our proposed blockchain-based Agri-Food supply chain system will strengthen the traceability information of both agricultural and food products. All the important aspects of a secure and efficient supply chain system are taken into consideration which are elaborated in the following sub-sections:

1) **Accountability**
By implementing blockchain in the proposed scheme, complete decentralization is achieved. It ensures the accountability of all actions by analyzing the logs. In this regard, the proposed solution allows arbitrators to become a part of the system and analyze the logs in case of disputes. The regulators have access to the blockchain data and can retrieve the required information in order to provide proof for accountability. Additionally, malicious nodes cannot succeed in performing a malicious act as the nodes are protected with standard signature scheme and it is impossible for the nodes to deny their actions.
2) **Credibility**
Credibility of the system is analyzed on the basis of level of trust among the entities of the system. The proposed solution; therefore, contains reputation system which is responsible for maintaining trust between the entities like, product owner, purchaser, LC, etc. Moreover, leveraging from the inherent properties of blockchain, the proposed solution is proved to be credible and secure. Hackers cannot hack the proposed system as long as they occupy more than 51% of all nodes.
3) **Auditability**
The complete system is auditable by any legitimate user of the system. It provides traceable smart contracts to track the transactions and events occurred. Blockchain provides the benefits like transparency, immutability and traceability. It ensures that the transactions are unforgeable.
4) **Autonomy**
All transactions and data exchanges in the proposed solution take place using smart contracts and prevent any kind of external interference. Hence, it ensures autonomy and security in trustful environment. Moreover, consensus based verification of blocks is also viewed as an autonomous property of blockchain-based solutions.
5) **Authenticity**
All the entities in the proposed solution are authenticated before performing the transaction. The authentication process ensures that certain functions are executed by authorized entities of the system only. Consequently, it also ensures the resistance to man-in-the-middle attacks.

VI. CONCLUSION AND FUTURE WORKS

Using blockchain, supply chain industry has gained numerous benefits to grow and move towards decentralization and achieve a trustless environment for all processes. However, despite the trustless nature of blockchain, it is hard to fully maintain trust between the seller and buyer of the product. This is because the entities may act maliciously and the buyer can doubt their credibility. Moreover, supply chain involves multiple processes and sub-processes that need to be carried out in a decentralized manner in order to achieve traceability, accountability and security. In this paper, we have proposed an end to end solution for blockchain-based Agri-Food supply chain. We have provided detailed information of proposed solution in terms of traceability, trading, delivery and reputation. We have evaluated and carefully analyzed the performance of smart contracts in order to ensure that the proposed solution is efficient and robust. The reputation system is proposed to maintain the credibility of the Agri-Food supply chain entities and quality ratings of the products. Moreover, it also maintains the immutability and integrity of the transactions as these transactions are based on blockchain. We have provided algorithms and discussed smart contracts in detail. Simulation are performed and results show that our system requires certain amount of gas for deploying and executing smart contracts.

To date, blockchain-based systems still face challenges related to their practical implementation. In future, we plan to integrate refund and return mechanisms in Agri-Food products trading. Similarly, the reputation system stores reviews from end consumers which can be biased or fake. In this regard, we plan to integrate fake review detection system that will facilitate the reputation system in detecting the false reviews from the end consumers. Moreover, security analyses that will focus on attacks against reputation system will also be considered.

REFERENCES

[1] M. Tripoli and J. Schmidhuber, "Emerging opportunities for the application of blockchain in the agri-food industry," FAO and ICTSD, Rome, Italy, Tech. Rep. CC BY-NC-SA 3, 2018.

[2] K. Malhotra, L. P. Ritzman, and S. K. Srivastava, *Operations Management: Processes and Supply Chain*. London, U.K.: Pearson, 2019.

[3] J. F. Galvez, J. C. Mejuto, and J. Simal-Gandara, "Future challenges on the use of blockchain for food traceability analysis," *TrAC Trends Anal. Chem.*, vol. 107, pp. 222–232, Oct. 2018.

[4] A. M. Turri, R. J. Smith, and S. W. Kopp, "Privacy and RFID technology: A review of regulatory efforts," *J. Consum. Affairs*, vol. 51, no. 2, pp. 329–354, Jul. 2017.

[5] A. Schaub, R. Bazin, O. Hasan, and L. Brunie, "A trustless privacy-preserving reputation system," in *Proc. IFIP Int. Conf. ICT Syst. Secur. Privacy Protection*. Cham, Switzerland: Springer, 2016, pp. 398–411.

[6] D. K. C. Lee, Ed., *Handbook of Digital Currency: Bitcoin, Innovation, Financial Instruments, and Big Data*. New York, NY, USA: Academic, 2015.

[7] M. Andoni, V. Robu, D. Flynn, S. Abram, D. Geach, D. Jenkins, P. McCallum, and A. Peacock, "Blockchain technology in the energy sector: A systematic review of challenges and opportunities," *Renew. Sustain. Energy Rev.*, vol. 100, pp. 143–174, Feb. 2019.

[8] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with IoT. Challenges and opportunities," *Future Gener. Comput. Syst.*, vol. 88, pp. 173–190, Nov. 2018.

[9] K. Salah, N. Nizamuddin, R. Jayaraman, and M. Omar, "Blockchain-based soybean traceability in agricultural supply chain," *IEEE Access*, vol. 7, pp. 73295–73305, 2019.

[10] J. Hao, Y. Sun, and H. Luo, "A safe and efficient storage scheme based on blockchain and IPFS for agricultural products tracking," *J. Comput.*, vol. 29, no. 6, pp. 158–167, 2018.

[11] S. Wang, X. Tang, Y. Zhang, and J. Chen, "Auditable protocols for fair payment and physical asset delivery based on smart contracts," *IEEE Access*, vol. 7, pp. 109439–109453, 2019.

[12] A. Shahid, U. Sarfraz, M. W. Malik, M. S. Iftikhar, A. Jamal, and N. Javaid, "Blockchain-based reputation system in agri-food supply chain," in *Proc. 34th Int. Conf. Adv. Inf. Netw. Appl. (AINA)*. Caserta, Italy: Univ. Campania Luigi Vanvitelli, Apr. 2020, pp. 12–21.

[13] M. P. Caro, M. S. Ali, M. Vecchio, and R. Giaffreda, "Blockchain-based traceability in agri-food supply chain management: A practical implementation," in *Proc. IoT Vertical Topical Summit Agricult.-Tuscany (IOT Tuscany)*, May 2018, pp. 1–4.

[14] F. Tian, "A supply chain traceability system for food safety based on HACCP, blockchain & Internet of Things," in *Proc. Int. Conf. Service Syst. Service Manage.*, 2017, pp. 1–6.

[15] Z. Li, H. Wu, B. King, Z. B. Miled, J. Wassick, and J. Tazelaar, "A hybrid blockchain ledger for supply chain visibility," in *Proc. 17th Int. Symp. Parallel Distrib. Comput. (ISPDC)*, Jun. 2018, pp. 118–125.

[16] M. Nakasumi, "Information sharing for supply chain management based on block chain technology," in *Proc. IEEE 19th Conf. Bus. Informat. (CBI)*, vol. 1, Jul. 2017, pp. 140–149.

[17] Q. Lu and X. Xu, "Adaptable blockchain-based systems: A case study for product traceability," *IEEE Softw.*, vol. 34, no. 6, pp. 21–27, Nov. 2017.

[18] D. Tse, B. Zhang, Y. Yang, C. Cheng, and H. Mu, "Blockchain application in food supply information security," in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manage. (IEEM)*, Dec. 2017, pp. 1357–1361.

[19] Y.-P. Lin, J. Petway, J. Anthony, H. Mukhtar, S.-W. Liao, C.-F. Chou, and Y.-F. Ho, "Blockchain: The evolutionary next step for ICT E-agriculture," *Environments*, vol. 4, no. 3, p. 50, 2017.

[20] R. AlTawy, M. El Sheikh, A. M. Youssef, and G. Gong, "Lelantos: A blockchain-based anonymous physical delivery system," in *Proc. 15th Annu. Conf. Privacy, Secur. Trust (PST)*, Aug. 2017, pp. 15–1509.

[21] K. Toyoda, P. T. Mathiopoulos, I. Sasase, and T. Ohtsuki, "A novel blockchain-based product ownership management system (POMS) for anti-counterfeits in the post supply chain," *IEEE Access*, vol. 5, pp. 17465–17477, 2017.

[22] H. R. Hasan and K. Salah, "Blockchain-based proof of delivery of physical assets with single and multiple transporters," *IEEE Access*, vol. 6, pp. 46781–46793, 2018.

[23] S. Wang, Y. Zhang, and Y. Zhang, "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems," *IEEE Access*, vol. 6, pp. 38437–38450, 2018.

[24] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decis. Support Syst.*, vol. 43, no. 2, pp. 618–644, Mar. 2007.

[25] Z. Yang, K. Zheng, K. Yang, and V. C. M. Leung, "A blockchain-based reputation system for data credibility assessment in vehicular networks," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–5.

[26] Y. Chen, H. Li, K. Li, and J. Zhang, "An improved P2P file system scheme based on IPFS and blockchain," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 2652–2657.

[27] Rinkeby. *Ethereum Testnet*. Accessed: Dec. 6, 2019. [Online]. Available: https://www.rinkeby.io/

[28] *Welcome to Remix Documentation! Remix, Ethereum-IDE 1 Documentation*. Accessed: Dec. 6, 2019. [Online]. Available: https://remix-ide.readthedocs.io/

[29] Truffle Suite. *Ganache: Ganache Quickstart: Documentation*. Accessed: Dec. 6, 2019. [Online]. Available: https://www.trufflesuite.com/docs/ganache/quickstart

[30] *MetaMask*. Accessed: Dec. 6, 2019. [Online]. Available: https://medium.com/metamask

[31] Y. Lu, "The blockchain: State-of-the-art and research challenges," *J. Ind. Inf. Integr.*, vol. 15, pp. 80–90, Sep. 2019.

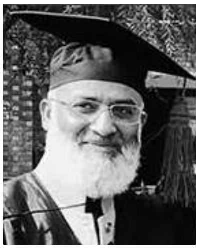
[32] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2016, pp. 254–269.

[33] Y. P. Tsang, K. L. Choy, C. H. Wu, G. T. S. Ho, and H. Y. Lam, "Blockchain-driven IoT for food traceability with an integrated consensus mechanism," *IEEE Access*, vol. 7, pp. 129000–129017, 2019.

[34] K. Behnke and M. F. W. H. A. Janssen, "Boundary conditions for traceability in food supply chains using blockchain technology," *Int. J. Inf. Manage.*, vol. 52, Jun. 2020, Art. no. 101969.

AFFAF SHAHID was born in Rawalpindi, Pakistan. She received the bachelor's degree in computer science from COMSATS University Islamabad. She is currently pursuing the master's degree in software engineering, under the supervision of Dr. Nadeem Javaid. She is a member of ComSens (Communication over Sensors) Research Group, Department of Computer Science, COMSATS University Islamabad. Her research interests include blockchain in the IoT, the IoV, and SCM.

AHMAD ALMOGREN (Senior Member, IEEE) received the Ph.D. degree in computer science from Southern Methodist University, Dallas, TX, USA, in 2002. He is currently a Professor of computer science with the Department, College of Computer and Information Sciences (CCIS), King Saud University (KSU), Riyadh, Saudi Arabia. He is also the Director of the Cyber Security Chair, CCIS, KSU. He has served as the Dean of the College of Computer and Information Sciences and the Head of the Academic Accreditation Council, Al Yamamah University. His research interests include mobile-pervasive computing and cyber security. He also served as the General Chair for the IEEE Smart World Symposium and a Technical Program Committee Member for numerous international conferences/workshops, such as IEEE CCNC, ACM BodyNets, and IEEE HPCC.



NADEEM JAVAID (Senior Member, IEEE) received the bachelor's degree in computer science from Gomal University, Dera Ismail Khan, Pakistan, in 1995, the master's degree in electronics from Quaid-i-Azam University, Islamabad, Pakistan, in 1999, and the Ph.D. degree from the University of Paris-Est, France, in 2010. He is currently an Associate Professor and the Founding Director of the Communications over Sensors (ComSens) Research Laboratory, Department of Computer Science, COMSATS University Islamabad, Islamabad. He has supervised 120 master's and 16 Ph.D. theses. He has authored over 900 articles in technical journals and international conferences. His research interests include energy optimization in smart/micro grids, wireless sensor networks, big data analytics in smart grids, and blockchain in WSNs, smart grids. He was a recipient of the Best University Teacher Award from the Higher Education Commission of Pakistan, in 2016, and the Research Productivity Award from the Pakistan Council for Science and Technology, in 2017. He is also an Associate Editor of IEEE ACCESS, an Editor of the *International Journal of Space-Based and Situated Computing*, and an Editor of the Sustainable Cities and Society.

FAHAD AHMAD AL-ZAHRANI received the B.Sc. degree in electrical and computer engineering from Umm Al-Qura University, Makkah, Saudi Arabia, in 1996, the M.S. degree in computer engineering from the Florida Institute of Technology, in 2000, and the Ph.D. degree in computer engineering from Colorado State University, in 2005. From 2011 to 2016, he was the IT Dean with Umm Al-Qura University and has had several other responsibilities thereafter. He has taught several computer network courses and supervised related research projects. He is currently an Associate Professor with the Computer Engineering Department, Umm Al-Qura University. His research interests include high-speed network protocols, sensor networks, optical networks, performance evaluation, IoT, and blockchain architecture and performance analysis. He is a member of the International Society for Optical Engineering and the Optical Society of America.

MANSOUR ZUAIR received the B.S. degree in computer engineering from King Saud University and the M.S. and Ph.D. degrees in computer engineering from Syracuse University. He has served as the CEN Chairman, from 2003 to 2006, and the Vice Dean, from 2009 to 2015. He has been the Dean, since 2016. He is currently an Associate Professor with the Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. His research interests include computer architecture, computer networks, and signal processing.

MASOOM ALAM received the Ph.D. degree in computer science from the University of Innsbruck, Austria, in November 2007, funded by HEC, Pakistan. Prior to that, he was a Lecturer with IMSciences. He has a total of 10 years' experience in teaching, system administration, and research and development. He is able to work on own initiatives as well as part a team. Having good leadership skills in research and development, he has been leading a Research Group with over 35 graduates and bachelor's students.