AgriOnBlock: Secured data harvesting for agriculture sector using blockchain technology

Article *in* ICT Express · July 2021

DOI: 10.1016/j.icte.2021.07.003

Bela Shrimali
Nirma University

ARTICLE IN PRESS

The Korean Institute of Communications and Information Sciences

Available online at www.sciencedirect.com

ScienceDirect

ICT Express xxx (xxxx) xxx

www.elsevier.com/locate/icte

AgriOnBlock: Secured data harvesting for agriculture sector using blockchain technology

Hiren Patel[a], Bela Shrimali[b],*

[a] Vidush Somany Institute of Technology and Research, Gujarat, India
[b] LDRP Institute of Technology and Research, Gujarat, India

Received 2 March 2021; accepted 13 July 2021
Available online xxxx

Abstract

The existing agriculture system is having several components such as supply chain management, crop insurance, the shipment of goods, and it involves numerous untrusted stakeholders such as users (farmers, retailers, customers, wholesalers, etc.), agencies (bank, insurance company) and regulators (assessor, surveyors). Due to the active engrossment of the middle man (mostly human beings or agencies operated by human beings) some key issues are raised such as transparency, timeliness, traceability, security, and immutability resulting in financial loss, crop contamination, and spoilage. Secure distributed public ledger technology and decentralized computing paradigm make Blockchain an appropriate alternative to resolve these issues and to achieve profitability & trust for all its stakeholders. In this research, we intend to propose a Blockchain-based mechanism viz. AgriOnBlock which would address the issues mentioned in the agriculture sector by connecting various stakeholders through the usage of IoT devices and smart contracts in Ethereum. We further discuss implications, constraints (methodological, awareness related, regulatory, etc.), and potential for actual adoption of AgriOnBlock.

Keywords: Agriculture; Blockchain technology; Crop insurance; Security; Smart contract; Supply chain management

1. Introduction

According to the UN Food and Agriculture Organization reports of 2012, 2.5 billion people in developing countries obtained their livelihood from agriculture that was over a third of the world's population [1] and the number kept on growing since then. In developing economics like India, farmers without sufficient technical knowledge, financial resources, and business skills struggle to receive a proper monetary gain from their crops. Their part of the profit goes to intermediaries due to lack of direct communication platform between two end parties viz. farmers and consumers. Due to enormous human intervention at every stage of supply chain management and on few occasions due to corrupt intermediaries, farmers do not get what actually they are entitled to. Along with supply chain management, an insurance claim is also an important module in the agriculture sector wherein receiving the insurance amount from legitimate agencies is a very tedious and cumbersome process for a farmer. There is a need of a time to make use of technology to reduce time consumption and to ease the process for the betterment of genuine stakeholders. Blockchain is one of such technologies which may help to integrate various non-trusting entities on a single platform with public verifiability and adequate security aspects. As per NIST [2], Blockchains are immutable distributed digital ledgers implemented without a central repository/authority. The transactions stored on Blockchain are immutable that is they cannot be altered or deleted, and every node of the network is having a complete set of a chain containing all the transactions till date. Hence, the transactions and authenticity can be publicly verifiable. Hence, Blockchain is used to improve safety, efficiency, and accountability at every stage of the process [3]. Blockchain is a chain of multiple blocks where each block contains a set of transactions and each block is linked with other blocks with a cryptographically secure hash code. The blocks are linked together in such a way that making a change in one block would require changing the entire chain, which is theoretically possible but practically infeasible. Appropriate hard work (sometimes in form of computational) is required to add a block into a chain

* Corresponding author.
E-mail addresses: hbpatel1976@gmail.com (H. Patel), bela.shrimali@gmail.com (B. Shrimali).

Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2021.07.003

by an authorized node of the network (sometimes known as a miner), and such miners get mining rewards, some gain in form of finance or otherwise, upon successfully adding the block into the chain. The idea of Blockchain was introduced for a cryptocurrency called Bitcoin by Satoshi Nakamoto [4]. The success of Bitcoin leads to enhancement to the application domain and introduced a programmable Blockchain via the use of smart contracts [5]. Smart contracts are self-executing contracts or codes with the terms of the agreement between buyer and seller being directly written into lines of code. The code and the agreements contained therein exist across a distributed, decentralized Blockchain network. Blockchain is adopted by many of the organizations and in the many domains viz. healthcare [6], supply chain [7], transportation [8] and many more. Agriculture is also one of the prominent domains which can have utilization of Blockchain technology to address the issues being face as mentioned above. In this paper, we explore three potential scenarios viz. (1) supply chain traceability (2) insurance claims, and (3) shipments covering many stakeholders like farmers, retailers, distributors and end customers. To address these scenarios, we present a smart contract-based Blockchain mechanism called AgriOnBlock which is permissioned Blockchain with a decentralized traceability function. Specifically, we intend to implement AgriOnBlock on the Ethereum [9], which is a global, open-source platform for decentralized applications. AgriOnBlock guarantees transparent and auditable asset traceability. To assess the feasibility of the proposed solution, we discuss different use-cases including Shipment from Warehouse to Retailer, Insurance Claim, and Supply Chain Management. We further present the templates of the smart contracts with these use cases. The rest of this paper is organized as follows. Section 2 summarizes the background of Blockchain technology and discusses various issues in the agriculture sector and subsequent opportunities through Blockchain. Section 3 describes the state of the art/related work in the domain. In Section 4, we present our proposal viz. AgriOnBlock. Section 5 discusses the tools and technology used to implement the proposed method and lastly, in Section 6, we conclude the paper followed by future prospective directions in this regard.

2. Background

In this section, we intend to first explore various issues currently being faced by the farmers in the agriculture sector and subsequently we depict how Blockchain technology could be useful in addressing the issues.

2.1. Issues in agriculture sector

Agriculture is one of the major contributors to GDP to any agriculture-based economics. Particularly India, with the production of agriculture activity of $375.61 billion is 2nd larger producer of agriculture products. India accounts for 7.39% of total global agricultural output. The contribution of the Agriculture sector in the Indian economy is much higher than the world's average (6.4%) [10]. In spite of all these facts and along with fertile land with water, the situation of a common farmer in India is not very good and there are various reasons behind it. In this section, we discuss the reasons for the same.

1. Too many intermediaries, information asymmetry: In the agriculture supply chain, crop moves from farm to warehouse to wholesaler/distributor to retailer to consumer. The price of the crops gets changed at each step. Every intermediary adds its own profit percentage to the original price resulting in the higher price of the crop for the end-user whereas the farmer still gets a very small part of the final amount. Also, due to the lack of variable-natured technology, there remains asymmetry in information/data shared among the stakeholders.
2. Lack of transparency and traceability in the agriculture supply chain: From the cultivation, treatment, and harvesting in the field to transportation, storage, quality control to production and then to the customer, the details are not maintained in any uniform way. All details are tracked and made available on the Blockchain system that the customer can verify.
3. Reducing the transaction delay/charge (by avoiding or reducing the role of intermediaries): In developing economies like India, where poor farmers are made to pay high premiums, Blockchain solution cuts out the middle person, resulting in higher profit gain on the part farmers, reduction in processing time & cost and ease in product and transaction chain management in agriculture.
4. Misusing and tempering of records: In the current scenario, the price of a crop/product varies every day mainly due to global market change and the quality of the product. One may try to sell a low-quality product at high prices or vice-versa with manual tempering. Technology can be used to address many issues in the agriculture sector. One of such technologies which have emerged for the last few years is Blockchain technology. In the next subsection, we describe how Blockchain technology can be utilized to address the issues raised above.

2.2. Opportunities through blockchain

Blockchain technology has gained popularity due to its distributive nature and highly secure mechanism to make its data tempered proof. By applying Blockchain to agriculture one may reduce fraud and cheating among untrusting partners. Following are the few of the opportunities through Blockchain to enhance efficiency in the agriculture sector:

- Transparency in supply chain management: Once the transaction is stored on the chain, it cannot be altered. However, all the transactions can be viewed by all the legitimate stakeholders of the system. Transactions that may contain data like crop origin, shipment timestamp, price, etc. cannot be modified once they are stored on Blockchain. Thus, every stakeholder transparently tracks the location and quality of the crop which is also known as public verifiability.

- Fair prices of crops and payment facilities to stakeholders: Every stakeholder transparently views the original price of the crop. So, farmers, retailers, and wholesalers get a fair price for their crops. Alteration in the prices is not possible once it is written on Blockchain. Unstipulated price reduction or increase will be impossible resulting in the rare possibility of any fraud.
- Easy traceability and auditability: Starting from farmer & farm information to crop and transportation, all the information is digitally linked to the crops within the Blockchain; every stakeholder covering farmers to end-customers/consumers can explore everything by back-tracking the entire supply chain cycle. The food supply chain based on the Blockchain can help different stakeholders to access information related to the food's quality at every stage. As Blockchain brings transparency to the food supply chain network, it will be easier to figure out when and how food has been contaminated.
- Tracking and payments of subsidies: In case of the loss during a weather crisis, farmers can quickly apply for the crop insurance claim amount through the Blockchain. The transparent and immutable behavior of the Blockchain will enable insurance companies and other authorized parties can access the data provided by farmers easily. They can directly query the Blockchain to fetch the required information with the help of smart contracts. After the insurance claim request is approved, farmers will automatically get the requested amount in their respective wallets. Also, during the whole process farmers can trace the activity of approval of subsidy. Thus, through Blockchain farmers can get compensation seamlessly and quickly.

3. Related work

In recent years, enormous research is being carried out around Blockchain technology to make its conceivable use with other technologies such as IoT, machine learning, and wireless sensor networks, for practical adaptability for various applications. Agriculture is identified as one of the prospective areas to implement Blockchain with IoT to facilitate different processes like supply chain, insurance claims, land registrations/buy/sale, etc.

In this section we present a brief survey on current research on Blockchain in agriculture domain-containing modules such as supply chain, insurance claiming, and practical implementation of smart contract [3,11–21].

Caroet et al.[17] introduce AgriBlockIoT, a Blockchain-based application for the AgriFood supply chain. They converse the different use cases covering the main activities of agriculture. They also confer two possible implementation platforms viz. Ethereum [9] and Hyperledger Sawtooth [9] and claim traceability. They also evaluate and compare the performance of both the deployments, in terms of latency, CPU, and network usage. It has been mentioned in their work that the performance of their proposed work got influenced by the underlying hardware. Andreas Kamilaris et al. [12] study the impact of Blockchain technology in agriculture and food supply chain and discuss overall implications, challenges, and prospective constraints. They have also highlighted many challenges including technical aspects, education, policies, and regulatory frameworks. Kaijun et al. [22] propose agricultural supply chain system based public Blockchain on double chain architecture and present the dual chain structure and its storage mode, resource rent-seeking, matching mechanism, and consensus algorithm. Their results show that the chain of agricultural supply chain based double chain structure can take into account the openness and security of transaction information and the privacy of enterprise information, can self-adaptively complete rent-seeking and matching of resources, and greatly enhance the credibility of the public service platform and the overall efficiency of the system. Kaijun et al. [22] propose a trusted, self-organized, open, and ecological food traceability system that relied on Blockchain and Internet of Things (IoT) technologies covering all non-trusted stakeholders. They use IoT devices to replace manual recording and verification to reduce the human intervention to the system effectively. However, they do not discuss any particular smart contracts for automated warning codes in the system. Henry Kim et al. [3] discuss applications of Blockchain across the agricultural sector covering provenance tracking or traceability across the various stages of the global food supply chain ensures food safety both for direct consumers as well as for a global community vulnerable to a food-born pandemic. They further explore possible smart contracts and chain of custody records to mitigate instances of food fraud and identify untrustworthy middlemen and business practices that exploit both independent farmers and cooperatives. They present AgriLedger with pilot programs in Kenya, Myanmar, and Papua New Guinea. Yadav and Singh [23] discuss the recent trends about Blockchain research covering man security and privacy mechanisms in agriculture and subsequently provides future research directions. Pouyan et al. [24] present a model to increase trust among agricultural supply chain parties to guarantee food quality, safety, and sustainability from a supply chain management perspective. They also study Blockchain information systems for real-time agricultural food traceability. Special emphasis has been placed on the roles of the incorporation of the IoT in Blockchain-based solutions. However, the authors have not discussed any implementation or prototype. Demestichas et al. [25] discuss relevant existing commercial applications, challenges, and future prospects of the application of blockchain technologies in the agri-food supply chain. With their extensive literature survey, they show that Blockchains can advantageously help to achieve traceability by irreversibly and immutably storing data. It creates a unique level of credibility that contributes to a more sustainable food industry. Hang et al. [26] propose a blockchain-based fish farm platform to guarantee agriculture data integrity. They designed platform for preserving large amounts of agriculture data securely. They introduced different processes in the fish farm using the smart contract to reduce the risk of error or manipulation. Thus, they use smart contracts to automate the data processing in the fish farm and

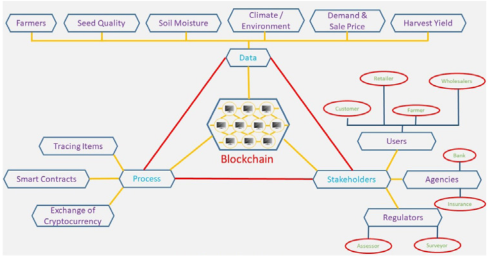


Fig. 1. Entities in Agriculture ecosystem.

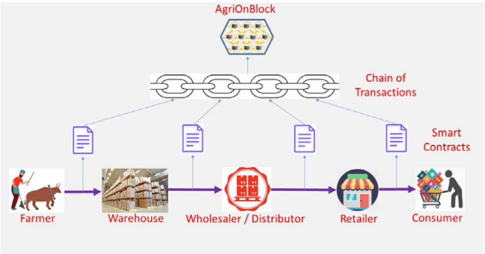


Fig. 2. Physical and digital flow in AgriOnBlock.

implemented a proof of concept that integrates a legacy smart fish farm system with the Hyperledger Fabric.

Moreover, there is much similar work carried out which discusses the efficient use of a smart contract for other applications too. Hasan and Salah [27] discuss the decentralized proof of delivery solution for digital assets. They use key features of blockchain and Ethereum smart contracts to provide immutable and tamper-proof logs, accountability, and traceability. They use Ethereum smart contracts to control all transactions and interactions for the sale and download of digital items. They demonstrate, implement, and test the full functionalities of smart contracts. Almadhoun et al. [28] propose a system design and implementation of a Blockchain-based solution using Ethereum smart contracts for IoT devices authentication. They authenticate numerous IoT devices including fog nodes with interfacing with the Ethereum Blockchain network. They discuss the security and vulnerability of the use of Ethereum smart contracts for their application.

4. Our proposal

4.1. Key components/Stakeholder of the system

Though there are many different stakeholders in agriculture ecosystem, we have identified few of them, which we wish to implement on smart contracts with Blockchain platform such as Ethereum during the latter part of this research. Fig. 1 shows all the entities involved in the AgriOnBlock. Broadly these entities are classified among three categories viz.

1. data
2. process and
3. stakeholders

Data is the entity that retrieves data from various sources such as farmer, seed quality, soil moisture, climate/environmental data, demand and sale price of various crops/seeds, and harvest yield. The various processes incorporated in the ecosystem are classified among (a) item traceability (b) smart contract and (c) (crypto) currency exchange. Finally, various stakeholders involved in the overall agricultural domain are divided among

- users such as customer, retailer, farmer, and wholesalers
- agencies such as bank and insurance company and
- regulators such as assessors and surveyors. The entities' data, processes, and stakeholders interact among themselves through the smart contract-based Blockchain technology AgriOnBlock. All the transactions being made among these entries are recorded on the immutable ledger AgriOnBlock for various purposes. Between various entities/stakeholders of the agriculture ecosystem, data/documents flow physically as well as logically. The transactions are combined together to form a block of a chain. Fig. 2 depicts how such various participants interact with each other and how the data/transactions are combined together to form a chain. All such records are stored on the immutable ledger AgriOnBlock for future purposes. The transaction among the participants (for example between the farmer and warehouse or between the retailer and the consumer) are driven through the code written in the smart contracts built for the same. Smart contracts are basically computer codes written for two parties to interact with each other on predefined agreements without a third party.

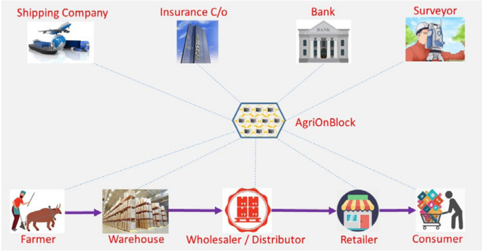


Fig. 3. Direct and indirect stakeholders of the system.

The stakeholders in the agriculture ecosystem can be divided among direct and indirect stakeholders based on the actual product/crop that they refer to. For example, farmers, warehouse, wholesaler/distributor, retailer, and consumer are direct stakeholders which deal with the crop/product directly. On the other side, shipping companies (which is more towards handling invoices), insurance companies, banks, and surveyors are indirect stakeholders of the system. Fig. 3 illustrates the communication among direct and indirect stakeholders of the system through the AgriOnBlock.

With this basic introduction on key components, participants, and stakeholders of the agriculture sector, in the next sub-section, we depict few case studies along with template code for the smart contracts.

Table 1
Acronyms and abbreviations.

Symbol	Details
$E_{PRRetailer}$	Private key of retailer
$E_{PURetailer}$	Public key of retailer
$ID_{Retailer}$	Unique identification of retailer
E_{PRBank}	Private key of bank
E_{PUBank}	Public key of bank
ID_{Bank}	Unique identification of bank
$E_{PRWarehouse}$	Private key of warehouse
$E_{PUWarehouse}$	Public key of warehouse
$ID_{Warehouse}$	Unique identification of warehouse
$E_{PRFarmer}$	Private key of farmer
$E_{PUFarmer}$	Public key of farmer
ID_{Farmer}	Unique identification of farmer
$E_{PRInsurance}$	Private key of insurance company
$E_{PUInsurance}$	Public key of insurance company
$ID_{Insurance}$	Unique identification of insurance company
$E_{PRSurveyor}$	Private key of surveyor
$E_{PUSurveyor}$	Public key of surveyor
$ID_{Surveyor}$	Unique identification of surveyor

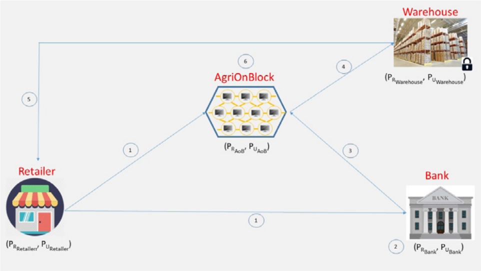

Warehouse
AgriOnBlock
Retailer
Bank
BANK
$(P_{R_{Warehouse}}, P_{U_{Warehouse}})$
$(P_{R_{AoB}}, P_{U_{AoB}})$
$(P_{R_{Retailer}}, P_{U_{Retailer}})$
$(P_{R_{Bank}}, P_{U_{Bank}})$
1
1
2
3
4
5
6

Fig. 4. Shipment from warehouse to retailer.

4.2. System architecture/design

In this subsection of the paper, we demonstrate three modules of the agricultural system viz. (a) Shipment from Warehouse to Retailer, (b) Insurance claim and (c) Supply Chain Management (Farmer, Retailer, and Distributor). Table 1 depicts the acronyms and abbreviations used in this section.

4.2.1. Shipment from warehouse to retailer

This module has been presented in Fig. 4. This module requires three stakeholders to interact with each other viz. (i) retailer (ii) bank and (iii) warehouse, through the Blockchain AgriOnBlock. The system flow of the same is shown in Fig. 5.

Following are the steps mentioned for the interactions among themselves. To avoid non-repudiation at later stage and to verify the authenticity of the sender, the transaction shall be encrypted using private key of the sender through public key cryptography algorithm such as RSA. Receiver would verify the authenticity of the sender using the public key of the sender.

1. When Retailer wants to retrieve some goods/product from warehouse, it communicates with the AgriOnBlock as well as Bank stating its identity, item code, item quantity along with its rate per unit.

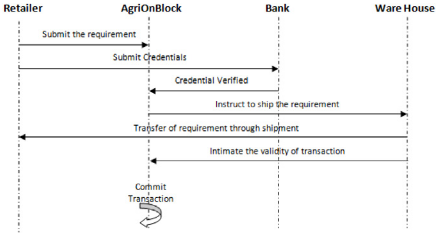


Fig. 5. System flow: Shipment from Warehouse to retailer.

2. Once the Retailer's credentials are verified and the Retailer has sufficient balance (in the bank) then, the required amount has been transferred from the account of the Retailer to that of AgriOnBlock.
3. Bank intimates AgriOnBlock regarding successful payment transaction.
4. Once the payment confirmation is received from Bank, AgriOnBlock instructs the Warehouse to transfer the required item in specified quantity to the Retailer, and the Warehouse transfers the required items to Retailer.
5. The transaction is added into the AgriOnBlock.

Following code demonstrates the smart contract template for the process of Shipment from Warehouse to Retailer

Algorithm 1 Smart Contract Template: Shipment from Warehouse to Retailer

```
SEND (i) $E_{PRRetailer}$ ($E_{PUAoB}$ ($ID_{Retailer}$, \$Amt, ItemCode, ItemQty)) FROM Retailer TO AgriOnBlock (ii) $E_{PRRetailer}$ ($E_{PUBank}$($ID_{Retailer}$, To: AgriOnBlock, \$Amt)) FROM Retailer TO Bank
IF (a) Retailer's credentials are verified AND (b) Retailer has sufficient balance THEN TRANSFER $Amt FROM BankAccountRetailer TO BankAccountWarehouse
IF amount transferred successfully THEN intimate the AgriOnBlock
IF confirmation received from Bank to AgriOnBlock THEN SEND $E_{PRAoB}$ ($E_{PUWarehouse}$(ItemCode, ItemQty) ) FROM AgriOnBlock TO Warehouse
SEND $E_{PRWarehouse}$($E_{PURetailer}$ (ItemCode, ItemQty) ) FROM Warehouse TO Retailer
Transaction is added into the AgriOnBlock
```

Sample contract in Solidity [25] has been illustrated beneath which shows how the amount is being transferred from Retailer to Warehouse.

```
    pragma solidity ĉompilerVersion;
    contract Warehouse2Retailer
    address public Retailer;
    mapping (address =¿ uint) public balances; –
    function send(address WareHouse, uint amount)
    public
    {
    –
    if (!(sender == Retailer && balances[Retailer] ¿ amount))
return;
```

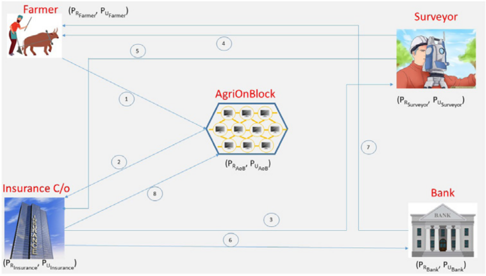

Farmer
$(P_{R_{Farmer}}, P_{U_{Farmer}})$
Surveyor
$(P_{R_{Surveyor}}, P_{U_{Surveyor}})$
AgriOnBlock
$(P_{R_{AoB}}, P_{U_{AoB}})$
Insurance C/o
$(P_{R_{Insurance}}, P_{U_{Insurance}})$
Bank
$(P_{R_{Bank}}, P_{U_{Bank}})$
1
2
3
4
5
6
7
8

Fig. 6. Insurance claim.

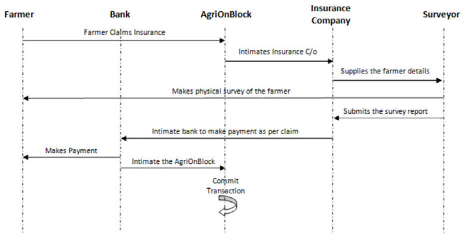


Fig. 7. System flow: Insurance claim.

balances[Retailer] -= amount; balances[WareHouse] += amount;

}–

4.2.2. *Insurance claim*

This module has been presented in Fig. 6. This module requires four stakeholders to interact with each other viz. (i) farmer (ii) insurance company (ii) surveyor and (iv) bank, through the Blockchain AgriOnBlock. Following are the steps mentioned for the interactions among themselves. The system flow of the same is shown in Fig. 7.

1. Farmer submits a claim for insurance to the Blockchain AgriOnBlock
2. After due verifications, Blockchain AgriOnBlock sends the details to the Insurance company.
3. Insurance company hands over the claim to Surveyor for physical verifications.
4. Surveyor physically makes survey by visiting the site.
5. Surveyor submits report to Insurance company with insurance amount to be settled.
6. After receiving the report from Surveyor, Insurance company intimates Bank to make payment to the Farmer.
7. Bank makes payment to Farmer.
8. Insurance Company intimates about the transaction to AgriOnBlock and the transaction is added to the Blockchain

Following code demonstrates the smart contract template for the process of Insurance claim

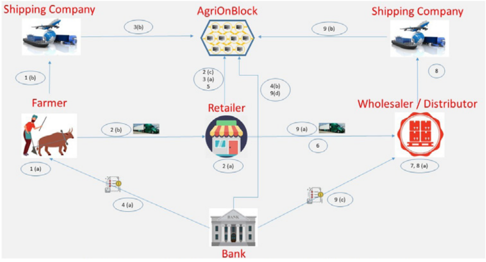


Fig. 8. Supply chain management (farmer, retailer, distributor).

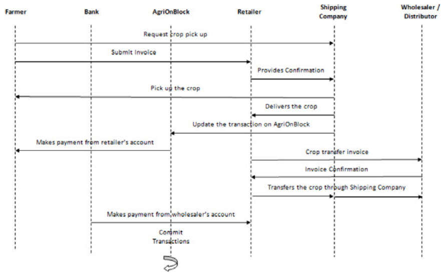


Fig. 9. System flow: Supply chain management (farmer, retailer, distributor).

Algorithm 2 Smart Contract Template: Insurance Claim

```
SEND $E_{PRFarmer}$ ($E_{PUAoB}$ ($ID_{Farmer}$, ClaimForInsurance)) FROM Farmer TO AgriOnBlock
SEND $E_{PRAoB}$ ($E_{PUInsurance}$ ($ID_{Farmer}$, ClaimForInsurance)) FROM AgriOnBlock TO InsuranceCompany
SEND $E_{PRInsurance}$ ($E_{PUSurveyor}$ ($ID_{Farmer}$, ClaimForInsurance)) FROM InsuranceCompany TO Surveyor
Surveyor physically makes survey by visiting the site.
SEND $E_{PRSurveyor}$ ($E_{PUInsurance}$ ($ID_{Farmer}$, ClaimForInsurance, \$Amt, OtherDetails)) FROM Surveyor TO InsuranceCompany
IF confirmation received from Surveyor THEN SEND $E_{PRInsurance}$ ($E_{PUBank}$($ID_{Farmer}$, \$Amt, Source Account: Insurance, Destination Account: Farmer)) FROM InsuranceCompany TO Bank
IF confirmation received from InsuranceCompany THEN SEND $E_{PRBank}$ ($E_{PUFarmer}$ ($ID_{Farmer}$, \$Amt, Source Account: Insurance, Destination Account: Farmer)) FROM Bank TO Farmer
InsuranceCompany intimates about the transaction to AgriOnBlock and the transaction is added to the Blockchain
```

4.2.3. *Supply chain management: Farmer, retailer, distributor*

This module and its system flow have been presented in Figs. 8 and 9. This module requires four stakeholders to interact with each other viz. (i) farmer (ii) retailer (ii) wholesaler and (iv) shipping companies, through the Blockchain AgriOnBlock. Following are the steps mentioned for the interactions among themselves. Here, the smart contract template is self-explanatory. Algorithm 3 demonstrates the smart contract template for the process of supply chain management and sample contract in Solidity [29] has been illustrated beneath which shows how the amount is being transferred from Insurance company to farmer based on the report of Surveyor.

Algorithm 3 Smart Contract Template: Supply Chain Management (Farmer, Retailer, Distributor)

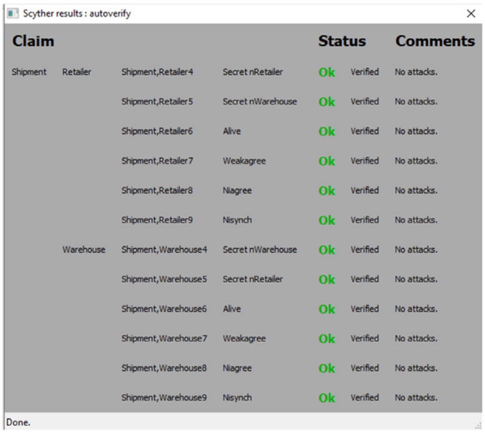
```
1: If the Farmer is not ready with a crop then stay on step 1 else go to step 2
2: (2.1) Farmer generates an invoice for the crop and it is sent to the Retailer (2.2) Request is sent to the Shipping company to pick up the crop
3: If the Retailer verifies the invoice received from the Farmer, then it sends confirmation Shipping company (and go to step 4) else go to error step (error code = 1)
4: (Upon receiving the invoice from the Farmer and confirmation from the Retailer) The Shipping company (4.1) picks up the crop from the Farmer and delivers it to the Retailer and (4.2) updates the transaction on the AgriOnBlock.
5: If the transaction updated by the Shipping company on AgriOnBlock is verified then (5.1) Smart contract between the Retailer and the Farmer gets executed to intimate the Bank to transfer the 'amount' from the Retailer to the Farmer and (5.2) the transaction is updated on the AgriOnBlock else go to error step (error code=2)
6: The Retailer generates an invoice for the crop and it is sent to the Wholesaler/Distributor.
7: If the Wholesaler/Distributor verifies of the invoice sent by the Retailer then (7.1) Wholesaler/Distributor sends a request to the Shipping company (7.2) the transaction is updated on the AgriOnBlock else go to error step (error code=3)
8: If the Shipping company verifies the invoice received from the Wholesaler/Distributor then (8.1) the Shipping company delivers the crop from the Retailer to the Wholesaler/Distributor and (8.2) the Shipping company updates the transaction on the AgriOnBlock else go to error step (error code=4)
9: Go to step 11.
10: Error : If error code = 1 then the Retailer intimates to the Farmer about the reasons for not accepting the invoice Else if error code = 2 then the Retailer and the Farmer are intimated about the denial of the Bank transaction Else if error code=3 then the Wholesaler/Distributor intimates the reasons of not accepting the invoice to the Retailer Else if error code=4 then the Shipping company intimates the reasons of not accepting the invoice to the Wholesaler/Distributor
11: STOP
```

```
Solidity pseudocode:
pragma solidity ĉompilerVersion;
contract insuranceClaim
{
address public insuranceCompany;
mapping (address =¿ uint) public balances;
–
function send(address farmer, uint sanctionedAmount) public
{
– if (!(sender == Surveyor && response[Surveyor] == "YES"')) return;
balances[insuranceCompany] -= sanctionedAmount;
balances[farmer] += sanctionedAmount;
}
–
}
```

4.3. Security analysis

In this subsection of our research, we discuss the security analysis of the proposed system. It elaborates how security goals like integrity, non-repudiation, confidentiality, and authentication can be achieved by providing sound authentication, access control, and data encryption schemes. We have used Scyther [30], the tool for the automatic verification of security protocols. It is one of the formal security parameters verification tools that is designed for automatic verification, falsification, and analysis of security protocols. Basic four claims of Scyther viz. Commit, Aliveness, Secret, and Nisynch use to verify the Impersonation attack, Authentication check, Confidentiality, and No replay attack respectively. We have made these claims for two processes: shipment and Insurance claim. For the shipment process, these security claims are checked between the warehouse and the retailer. Wherein, for the insurance claims process, the security goals/claims are checked for all participating users viz. (a) between the Farmer and the AgriOnBlock (b) between Insurance company and Surveyor (c) between Insurance company and Bank (d) between Bank and Farmer. As discussed, we mainly intend to achieve (i) Integrity (ii) Non-repudiation (iii) Confidentiality, and (iv) Authentication. We checked the security of both the shipment and insurance claim process using Scyther. Figs. 10, 11, 12, 13 and 14 justify that the security goals are achieved and maintained in both the process.

1. Integrity: It is about making sure that the data/information sent from one party to another has not been tempered. In our SPDL scripting, we use a hash function (such as SHA-256) to make sure that the data sent has not been altered during transportation. The hash code has been computed at the sender site and sent along with the original data, and recomputed at the receiver side. The recomputed hash code at the receiver side is compared with what has been sent from the sender, and if they both match, we conclude that nothing happened to our data and the integrity has been protected.
2. Non-repudiation: It is about restricting the entity for denial for the action that is performed. To achieve non-repudiation, we have used the private key of the sender to sign the transaction, so that later the sender cannot deny that the transaction was not made by it, as the possession of a private key is expected to be with the authorized entity only. In this way, we resolve the issue of non-repudiation.

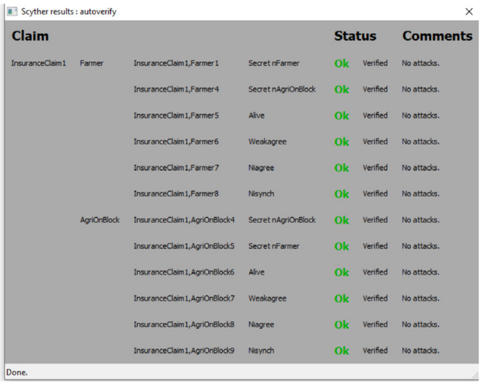


Scyther results : autoverify

Claim				Status		Comments
Shipment	Retailer	Shipment,Retailer4	Secret nRetailer	Ok	Verified	No attacks.
		Shipment,Retailer5	Secret nWarehouse	Ok	Verified	No attacks.
		Shipment,Retailer6	Alive	Ok	Verified	No attacks.
		Shipment,Retailer7	Weakagree	Ok	Verified	No attacks.
		Shipment,Retailer8	Niagree	Ok	Verified	No attacks.
		Shipment,Retailer9	Nisynch	Ok	Verified	No attacks.
	Warehouse	Shipment,Warehouse4	Secret nWarehouse	Ok	Verified	No attacks.
		Shipment,Warehouse5	Secret nRetailer	Ok	Verified	No attacks.
		Shipment,Warehouse6	Alive	Ok	Verified	No attacks.
		Shipment,Warehouse7	Weakagree	Ok	Verified	No attacks.
		Shipment,Warehouse8	Niagree	Ok	Verified	No attacks.
		Shipment,Warehouse9	Nisynch	Ok	Verified	No attacks.

Done.

Fig. 10. Verifying "Shipment Process" against attacks.



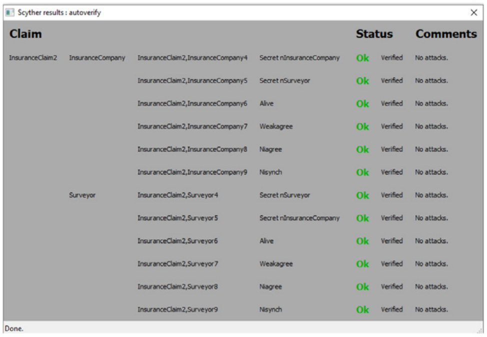
Scyther results : autoverify

Claim				Status		Comments
InsuranceClaim1	Farmer	InsuranceClaim1,Farmer1	Secret nFarmer	Ok	Verified	No attacks.
		InsuranceClaim1,Farmer4	Secret nAgriOnBlock	Ok	Verified	No attacks.
		InsuranceClaim1,Farmer5	Alive	Ok	Verified	No attacks.
		InsuranceClaim1,Farmer6	Weakagree	Ok	Verified	No attacks.
		InsuranceClaim1,Farmer7	Niagree	Ok	Verified	No attacks.
		InsuranceClaim1,Farmer8	Nisynch	Ok	Verified	No attacks.
	AgriOnBlock	InsuranceClaim1,AgriOnBlock4	Secret nAgriOnBlock	Ok	Verified	No attacks.
		InsuranceClaim1,AgriOnBlock5	Secret nFarmer	Ok	Verified	No attacks.
		InsuranceClaim1,AgriOnBlock6	Alive	Ok	Verified	No attacks.
		InsuranceClaim1,AgriOnBlock7	Weakagree	Ok	Verified	No attacks.
		InsuranceClaim1,AgriOnBlock8	Niagree	Ok	Verified	No attacks.
		InsuranceClaim1,AgriOnBlock9	Nisynch	Ok	Verified	No attacks.

Done.

Fig. 11. Verifying insurance claim communication between "Farmer and AgriOnBlock" against attacks.

3. Confidentiality: It is about making sure that the message being sent from the sender to the receiver is in encrypted form (ciphertext) and no entity other than the authenticated parties should be able to decipher the encrypted message. To achieve this, we have used symmetric key encryption (such as AES) to maintain confidentiality.
4. Authentication: It is about making sure about the originality of the sender/receiver and which can be achieved through the private key of the sender using asymmetric key cryptography. We propose to use algorithms such as RSA to achieve authentication by using the concept of the digital signature.

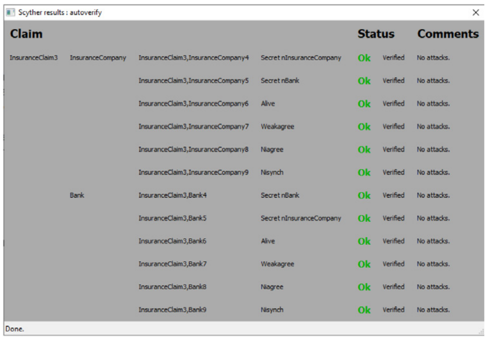


Scyther results : autoverify

Claim				Status		Comments
InsuranceClaim2	InsuranceCompany	InsuranceClaim2,InsuranceCompany4	Secret nInsuranceCompany	Ok	Verified	No attacks.
		InsuranceClaim2,InsuranceCompany5	Secret nSurveyor	Ok	Verified	No attacks.
		InsuranceClaim2,InsuranceCompany6	Alive	Ok	Verified	No attacks.
		InsuranceClaim2,InsuranceCompany7	Weakagree	Ok	Verified	No attacks.
		InsuranceClaim2,InsuranceCompany8	Niagree	Ok	Verified	No attacks.
		InsuranceClaim2,InsuranceCompany9	Nisynch	Ok	Verified	No attacks.
	Surveyor	InsuranceClaim2,Surveyor4	Secret nSurveyor	Ok	Verified	No attacks.
		InsuranceClaim2,Surveyor5	Secret nInsuranceCompany	Ok	Verified	No attacks.
		InsuranceClaim2,Surveyor6	Alive	Ok	Verified	No attacks.
		InsuranceClaim2,Surveyor7	Weakagree	Ok	Verified	No attacks.
		InsuranceClaim2,Surveyor8	Niagree	Ok	Verified	No attacks.
		InsuranceClaim2,Surveyor9	Nisynch	Ok	Verified	No attacks.

Done.

Fig. 12. Verifying insurance claim communication between "Insurance company and surveyor" against attacks.

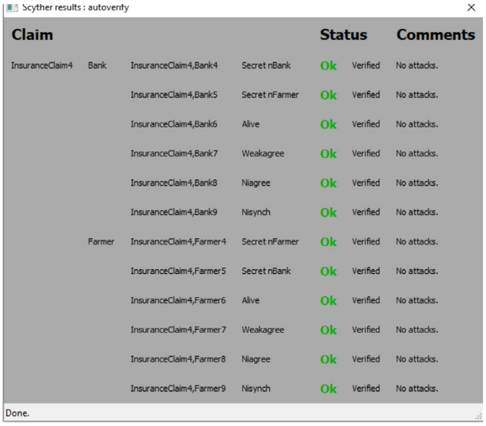


Scyther results : autoverify

Claim				Status		Comments
InsuranceClaim3	InsuranceCompany	InsuranceClaim3,InsuranceCompany4	Secret nInsuranceCompany	Ok	Verified	No attacks.
		InsuranceClaim3,InsuranceCompany5	Secret nBank	Ok	Verified	No attacks.
		InsuranceClaim3,InsuranceCompany6	Alive	Ok	Verified	No attacks.
		InsuranceClaim3,InsuranceCompany7	Weakagree	Ok	Verified	No attacks.
		InsuranceClaim3,InsuranceCompany8	Niagree	Ok	Verified	No attacks.
		InsuranceClaim3,InsuranceCompany9	Nisynch	Ok	Verified	No attacks.
	Bank	InsuranceClaim3,Bank4	Secret nBank	Ok	Verified	No attacks.
		InsuranceClaim3,Bank5	Secret nInsuranceCompany	Ok	Verified	No attacks.
		InsuranceClaim3,Bank6	Alive	Ok	Verified	No attacks.
		InsuranceClaim3,Bank7	Weakagree	Ok	Verified	No attacks.
		InsuranceClaim3,Bank8	Niagree	Ok	Verified	No attacks.
		InsuranceClaim3,Bank9	Nisynch	Ok	Verified	No attacks.

Done.

Fig. 13. Verifying insurance claim communication between "Insurance company and Bank" against attacks.



Scyther results : autoverify

Claim				Status		Comments
InsuranceClaim4	Bank	InsuranceClaim4,Bank4	Secret nBank	Ok	Verified	No attacks.
		InsuranceClaim4,Bank5	Secret nFarmer	Ok	Verified	No attacks.
		InsuranceClaim4,Bank6	Alive	Ok	Verified	No attacks.
		InsuranceClaim4,Bank7	Weakagree	Ok	Verified	No attacks.
		InsuranceClaim4,Bank8	Niagree	Ok	Verified	No attacks.
		InsuranceClaim4,Bank9	Nisynch	Ok	Verified	No attacks.
	Farmer	InsuranceClaim4,Farmer4	Secret nFarmer	Ok	Verified	No attacks.
		InsuranceClaim4,Farmer5	Secret nBank	Ok	Verified	No attacks.
		InsuranceClaim4,Farmer6	Alive	Ok	Verified	No attacks.
		InsuranceClaim4,Farmer7	Weakagree	Ok	Verified	No attacks.
		InsuranceClaim4,Farmer8	Niagree	Ok	Verified	No attacks.
		InsuranceClaim4,Farmer9	Nisynch	Ok	Verified	No attacks.

Done.

Fig. 14. Verifying insurance claim communication between "Bank and farmer" against attacks.

4.4. Cost analysis

Smart contracts are self-executable code similar to a class of other object-oriented programming languages like Java, Python, C++, and many more. But smart contracts are different in the nature of existence, which means, once they are deployed they are immutable i.e. the code written in it cannot be changed. There are many platforms to implement smart contracts out of which Remix Ethereum [9] one of the most popular ones among the research community. Miners try to mine the blocks in the chain and the miner who successfully mines a block gets a reward from the transaction fees. In the Ethereum environment, this transaction fee is a transaction executable cost specified in terms of gas. Every transaction executed by a smart contract in the Ethereum network costs some discrete amount of gas based on the number of Ethereum Virtual Machine (EVM) instructions. A gas limit is associated with every smart contract, which is the maximum amount of gas the owner is willing to pay for the transaction. So, Ethereum gas can be called a unit that measures the amount of computational effort that it requires to execute certain operations and every smart contract instruction has an associated gas consumption that relates the instruction to its storage or execution cost [31]. We consider that every single transaction requires at least 21,000 gas as mentioned in [9]. Based on this, for the sample smart contracts that we discussed in this section, we calculated an approximate cost for every contract using the EVM instruction costs mentioned in [9]. Our proposed smart contracts perform operations like mapping, jumps, condition checking, and arithmetic operations. So, considering all such operations, the approximately worst-case gas requirement is 30,00,000 gas.

5. Tools and technologies

Smart contracts are the primitive building blocks of code written to build the Blockchain and to run the Blockchain. Programming languages like Solidity, Golang, Javascript, C++, and Java are mostly used to start building smart contracts nowadays [32]. Smart contracts are immutable once they get deployed. They are also called a microservice on the web, it exists on Blockchain, and anyone can use it and execute a business logic written inside a program. Usually, the client running on the browser contains HTML, CSS, and Javascript as part of the frontend design. The client makes a request for database service to a web server that contains all the services/programs. All this relies on a central server. But in the case of Blockchain, initially, the web browser communicates with a front end and these web pages communicate with Blockchain. It will connect to the Ethereum node and access all the data on Blockchain and also interact and execute the code written as a smart contract. In this section, we discuss the tools and technologies being used for private and public Blockchain to build smart contracts. (a) Ethereum Ganache [33]: Ethereum [9] is an open-source, Blockchain-based, public, distributing computing environment that provides scripting functionality through smart contracts. It provides a Turing-complete Ethereum Virtual Machine (EVM), which executes smart contracts. Ethereum has a JavaScript library called web3.js which is used for creating the connection to the Blockchain. Ethereum smart contracts are executed within the Ethereum network. While Ethereum Ganache is a part of the Truffle Suite, a set of developer tools that allow users to create Blockchain locally and test smart contracts. It provides more features than Remix [34]. (b) Remix Ethereum: Remix [34] is one of the simplest tools provided on the official Ethereum site to test, debug and deploy a smart contract. Remix Ethereum environment provides JavaScript VM, injected web3 and web3 to execute a smart contract. Remix Integrated Development Environment (IDE) can be online, via a web browser or from a locally installed copy, or from Mist (the Ethereum Dapp browser). (c) Solidity programming: A solidity is a contract and object-oriented language especially developed for contract writing. It is a high-level language, which is derived from C++, Python, and JavaScript. The Solidity compiler compiles smart contract code into byte code that runs on Ethereum Virtual Machine (EVM).

6. Conclusion and future work

This research paper presents AgriOnBlock, a Blockchain-based smart contract technology for the agricultural domain to achieve transparency (e.g. financial), traceability (e.g. crop), trust (among non-trusting stakeholders), accountability (through non-repudiation using public key infrastructure). A prototype of AgriOnBlock has been presented which intends to circumvent/diminish financial loss, crop contamination, and spoilage resulting in profitability in less time. AgriOnBlock describes the main three entities in the agriculture domain viz. data, process, and stakeholders and elaborates them. In addition, we explored various processes such as (a) shipment from warehouse to retailer, (b) insurance claim, and (c) supply chain management (farmer, retailer, and distributor) with its security concerns. However we have assumed the access control policies (e.g. Role based, Attribute based etc.) are inbuilt on AgriOnBlock and the Blockchain takes care of them while retrieving data asked by a specific stakeholder. Imminent researcher may dig into these aspects to provide other options to the stake holders based on their requirements. In the later part of this research, we intend to implement the various modules on smart contracts with a Blockchain platform such as Ethereum. In the future, along with adding few more processes into the overall ecosystem of the agriculture sector, we would like to introduce (a) cryptocurrency for stakeholders involved in the agriculture domain and (b) land registration and (buyer/seller) traceability process.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] I.J. Winfield, Fao statistical yearbook 2012: World food and agriculture-edited by A. Prakash and M. Stigler, J. Fish Biol. 81 (6) (2012) 2095–2096.

[2] D. Yaga, P. Mell, N. Roby, K. Scarfone, Blockchain technology overview, 2019, arXiv preprint arXiv:1906.11078.

[3] H.M. Kim, M. Laskowski, Agriculture on the blockchain: Sustainable solutions for food, farmers, and financing, in: Supply Chain Revolution, Barrow Books, 2018.

[4] S. Nakamoto, Bitcoin: A Peer-To-Peer Electronic Cash System, Tech. Rep., Manubot, 2019.

[5] K. Lauslahti, J. Mattila, T. Seppala, Smart contracts–how will blockchain technology affect contractual practices? Etla Rep. (68) (2017).

[6] F. Angeletti, I. Chatzigiannakis, A. Vitaletti, Towards an architecture to guarantee both data privacy and utility in the first phases of digital clinical trials, Sensors 18 (12) (2018) 4175.

[7] R.K. Osei, M. Canavari, M. Hingley, An exploration into the opportunities for blockchain in the fresh produce supply chain, 2018.

[8] Y. Yuan, F.-Y. Wang, Towards blockchain-based intelligent transportation systems, in: 2016 IEEE 19th International Conference on Intelligent Transportation Systems, ITSC, IEEE, 2016, pp. 2663–2668.

[9] G. Wood, et al., Ethereum: A secure decentralised generalised transaction ledger, Ethereum Project Yellow Paper 151 (2014) (2014) 1–32.

[10] statisticTimes, 2020, http://statisticstimes.com/economy/sectorwise-gdp-contribution-of-india.php. (Accessed 2020).

[11] Y.-P. Lin, J.R. Petway, J. Anthony, H. Mukhtar, S.-W. Liao, C.-F. Chou, Y.-F. Ho, Blockchain: The evolutionary next step for ict e-agriculture, Environments 4 (3) (2017) 50.

[12] A. Kamilaris, A. Fonts, F.X. Prenafeta-Boldú, The rise of blockchain technology in agriculture and food supply chains, Trends Food Sci. Technol. 91 (2019) 640–652.

[13] M. Chinaka, Blockchain Technology–Applications in Improving Financial Inclusion in Developing Economies: Case Study for Small Scale Agriculture in Africa (Ph.D. thesis), Massachusetts Institute of Technology, 2016.

[14] S. Haveson, A. Lau, V. Wong, Protecting Farmers in Emerging Markets with Blockchain, SC Johnson College of Business, Cornell University, 2017.

[15] D.H. Beula, et al., Supply chain management in agriculture, Our Heritage 68 (4) (2020) 297–304.

[16] S.F. Papa, Use of blockchain technology in agribusiness: transparency and monitoring in agricultural trade, in: 2017 International Conference on Management Science and Management Innovation, MSMI 2017, Atlantis Press, 2017.

[17] M.P. Caro, M.S. Ali, M. Vecchio, R. Giaffreda, Blockchain-based traceability in agri-food supply chain management: A practical implementation, in: 2018 IoT Vertical and Topical Summit on Agriculture-Tuscany, IOT Tuscany, IEEE, 2018, pp. 1–4.

[18] M. Tripoli, J. Schmidhuber, Emerging Opportunities for the Application of Blockchain in the Agri-Food Industry, FAO and ICTSD, Rome and Geneva, 2018, Licence: CC BY-NC-SA 3.

[19] M. Schneider, Design and Prototypical Implementation of a Blockchain-Based System for the Agriculture Sector, vol. 10 (Master's thesis), Faculty of Business, Economics and Informatics, University of Zurich, 2017.

[20] D. Tse, B. Zhang, Y. Yang, C. Cheng, H. Mu, Blockchain application in food supply information security, in: 2017 IEEE International Conference on Industrial Engineering and Engineering Management, IEEM, IEEE, 2017, pp. 1357–1361.

[21] J. Lin, Z. Shen, A. Zhang, Y. Chai, Blockchain and iot based food traceability for smart agriculture, in: Proceedings of the 3rd International Conference on Crowd Science and Engineering, 2018, pp. 1–6.

[22] K. Leng, Y. Bi, L. Jing, H.-C. Fu, I. Van Nieuwenhuyse, Research on agricultural supply chain system with double chain architecture based on blockchain technology, Future Gener. Comput. Syst. 86 (2018) 641–649.

[23] V.S. Yadav, A. Singh, A systematic literature review of blockchain technology in agriculture, in: Proceedings of the International Conference on Industrial Engineering and Operations Management, 2019, pp. 973–981.

[24] S. Wingreen, R. Sharma, et al., A blockchain traceability information system for trust improvement in agricultural supply chain, 2019.

[25] K. Demestichas, N. Peppes, T. Alexakis, E. Adamopoulou, Blockchain in agriculture traceability systems: A review, Appl. Sci. 10 (12) (2020) 4113.

[26] L. Hang, I. Ullah, D.-H. Kim, A secure fish farm platform based on blockchain for agriculture data integrity, Comput. Electron. Agric. 170 (2020) 105251.

[27] H.R. Hasan, K. Salah, Proof of delivery of digital assets using blockchain and smart contracts, IEEE Access 6 (2018) 65439–65448.

[28] R. Almadhoun, M. Kadadha, M. Alhemeiri, M. Alshehhi, K. Salah, 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications, AICCSA, IEEE, 2018, pp. 1–8.

[29] C. Dannen, Introducing Ethereum and Solidity, vol. 1, Springer, 2017.

[30] URL https://people.cispa.io/cas.cremers/scyther/. (Accessed 2020).

[31] M. Marescotti, M. Blicha, A.E. Hyvärinen, S. Asadi, N. Sharygina, Computing exact worst-case gas consumption for smart contracts, in: International Symposium on Leveraging Applications of Formal Methods, Springer, 2018, pp. 450–465.

[32] URL https://medium.com/fleta-first-chain/6-popular-blockchain-programming-languages-used-for-building-smart-contracts-and-fleta-will-7b310f1a9e2. (Accessed 2020).

[33] W.-M. Lee, Esting smart contracts using ganache, in: Beginning Ethereum Smart Contracts Programming, Springer, 2019, pp. 147–167.

[34] URL https://www.sitepoint.com/remix-smart-contracts-ethereum-blockchain/. (Accessed 2020).