Journal of Information Security and Applications 54 (2020) 102554

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

Blockchain based secured information sharing protocol in supply chain management system with key distribution mechanism

Sanjeev Kumar Dwivedi, Ruhul Amin*, Satyanarayana Vollala

Department of Computer Science and Engineering, Dr. SPM International Institute of Information Technology, Naya Raipur, Naya Raipur-492 002, India

ARTICLE INFO

Keywords:
Pharmaceutical supply chain
Blockchain
Secure information sharing

ABSTRACT

The concept of Supply Chain Management (SCM) is very imperative while moving sensitive products from one entity to the next entity until it reaches to the end-users to avoid damage(s) in the product. In the traditional supply chain management system, several serious problems such as tampering of products, delay, and fraud, etc. exist. It also lacks proper authentication among the participants, data management as well as the integrity of the data. The blockchain mechanism is capable of solving the above-mentioned issues due to its important features such as decentralization, transparency, trust-less environment, anonymity, and immutability. This paper describes how the blockchain mechanism combines with the traditional pharmaceutical supply chain system and to achieve a better SCM system, we present a blockchain-based scheme for information sharing securely in the pharmaceutical supply chain system with smart contracts and consensus mechanism. The proposed scheme also provides a mechanism to distribute required cryptographic keys to all the participants securely using the smart contract technique. Further, transaction and block validation protocols have been designed in our protocol. The security analysis ensures that our protocol is robust and also achieves reasonable performance in terms of computation and communication overheads.

1. Introduction

Supply chain management (SCM) is the management of the entire production, flow of goods, data, and finances, and oversees the processes until it transforms it into final products or it reaches its final destination. It is the backbone of commerce. By managing the supply chain, the excess cost of delivering the product to the customer reduces [1]. For the smooth functioning of any industry, a proper, and well-managed supply chain is very important. Little chaos in the supply chain can disrupt the whole market and can cause huge financial loss to involve organizations. To detect the origin of counterfeit products which somehow have reached to the customer, a well maintained, and immutable supply chain is required [2].

A lot of work is going on to identify the products in the SCM process. Generally, it uses barcodes and radio frequency identification (RFID) tags. The tags used in the SCM process are low-cost and the capabilities of these tags are very limited. Therefore, the privacy and transparency of the products are a challenging task in the SCM system. Additionally, it also suffers from the well-known wireless attacks (reply attacks, eavesdropping, etc.). The term product privacy is very important in SCM. It means that only authenticated parties can access the private data of products, either it is stored in the private database or cloud so that intruders will not be able to forge the product data at any stage of the supply chain process. The interaction between the system and the participating entities requires a secure channel. Different authentication protocols in the different areas including supply chain such as in the healthcare-services [3,4], in the agricultural monitoring [5], in the field of wireless sensor networks (WSN) [6–8], in supply chain [9,10], and internet-of-things (IoT) domain [11,12] are suggested by the different authors.

1.1. Traditional pharmaceutical supply chain management (PSCM)

The pharmaceutical supply chain (PSC) operates very differently from other supply chains in which delivery of products are subjected to various kind of comprehensive regulations and rules. But, the rules must be strictly followed, and proper transparency along with trust are maintained among the different participants of the supply chain, as to deliver the product in due time, and proper condition [13]. The major stakeholders of the PSC network are raw material suppliers, manufacturers, warehouse owners, retail-

* Corresponding author.
E-mail addresses: amin_ruhul@live.com, sanjeevdwivedi131988@gmail.com (S.K. Dwivedi), amin_ruhul@live.com (R. Amin), ruhul@iiitnr.edu.in, satya4nitt@gmail.com (S. Vollala).

https://doi.org/10.1016/j.jisa.2020.102554
2214-2126/

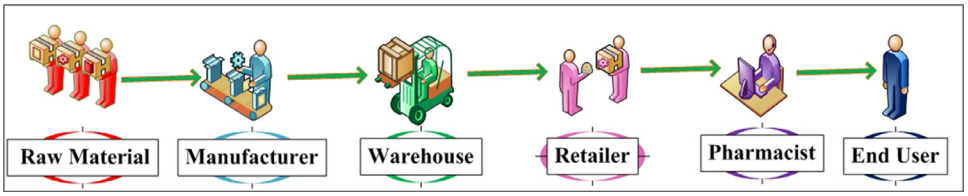

Fig. 1. Working mechanisms of traditional pharmaceutical supply chain without blockchain.

ers, pharmacists, and the end-users or the customers. The general flow of the PSC is shown in Fig. 1. The general working steps of traditional pharmaceutical supply chain management (PSCM) system is as follows:-

- In the first phase, the giant pharma manufacturers collect their raw materials from different suppliers to develop a product. A considerable amount of research and time is devoted to the proper synthesis of drugs.
- In the second phase, after the development of the product, the manufacturer sends it to the warehouse. The company using a single unit of manufacturing requires only one warehouse, whereas the one with multiple units of manufacturing stores its product in different warehouses (central or regional warehouses) depending upon the geographical conditions.
- In the third phase, the retailers receive the product and supply them to the next stakeholders in the chain pharmacists, or other similar entities such as hospitals, health care centers, and clinics. They usually order the products according to the demand, and needs.
- In the last phase, the pharmacists sell the drugs directly to the patients or the end-users.

1.2. Major shortcomings of existing pharmaceutical supply chain management

There are several drawbacks present in the existing PSCM system. Some of them are listed below [14,15]:

- **Lack of Transparency**: lack of visibility in PSCM is the main issue in the industry. Million-dollar have been invested to solve problems such as, to create a little more transparent supply chain, to identify each drug uniquely, and to solve the problem of counterfeit drugs. But, little success has been achieved in this arena.
- **Product Traceability**: In the traditional PSCM system, every stakeholder maintains its database that stores the information of a particular product. As a result, predictive monitoring of products is a very challenging task and it incurs extra time and delay to keep a manual record of every product. This problem can be easily solved by the adoption of a smart contracts-enabled blockchain-based PSCM system.
- **Lack of Trust**: The PSCM system involves a various number of participants until the product is delivered to the end-user. Maintaining the trust among the participants in such a complex and huge SCM is a challenging task. Consequently, it can also affect the smoothness of the supply chain process.
- **Shipping of Expiry Products**: Customers expect products, which are not expired but, sometimes the supply chain process consumes much amount of time which results in rejection of the drugs at the last stage thus, making the whole process futile.
- **Cold-Chain Shipping**: Most of the supply chain entities lack the equipment required to transport temperature-controlled drugs. And, the current pharma industry is facing upraise of this form of the drug, and thus, it causes huge loss of the drug products and money.
- **Counterfeit Products**: Due to the lack of transparency, and out-dated information-sharing mechanisms in the supply chain system, counterfeited products are delivered to the end-users which affect both the economy and users life. According to the Economic Cooperation and Development Organization report, half-trillion dollars of the global economy has been wasted per year in the SCM process.
- **Documentation and Regulatory Compliance**: The existing supply chain system involves paper-based trails for ownership-change, letters of credit, and complicated payment terms. Therefore, traditional supply chain contracts become complicated and out-dated. The blockchain-based supply chain system incorporates smart contracts that provide automation for the above-mentioned issues.

1.3. Pharmaceutical supply chain management with blockchain

All the phases of the supply chain are recorded in a blockchain network. Every new transaction performs in the network is stored in an immutable block, and time-stamped to keep track of the specific product in the end-to-end chain [16], and assures that the details present in the block are not tempered [17,18]. All the entities of the network can log in to the network, and verify the authenticity, and history of any drug. The general diagram of PSCM with blockchain is shown in Fig. 2.

Fig. 2. illustrates that how the different entities of the PSCM system interact and cooperate under the smart contracts enabled the blockchain network. Each participating entity submits its transaction (based on the completed activity) on the blockchain network. In the initial step, different suppliers deliver the raw materials to the manufacturer and submit their transactions on the blockchain network. This transaction includes the raw material name, quantity, and quality of material, location of the supplier, etc. Once the manufacturer gets the raw materials, smart contracts are automatically executed, and suitable action (e.g. money is directly transferred from manufacturer to supplier) is executed. Smart contracts are specific rules and regulations that every participating entity must follow [19]. Every participating entity can verify the submitted transaction on the blockchain network whether the transaction is legitimate or not? Similarly, on the next stage manufacturer has an interaction with the network, and so on.

Deployment of Blockchain network along with smart contracts or similar technologies would greatly help to keep proper track of the items, and triggers the payment process automatically, when necessary. Smart contracts are simple programs, which specify the conditions for validation of a particular transaction [16,20].

1.4. Major benefits of PSCM using blockchain

The blockchain-based PSCM system has several benefits over the traditional PSCM system. It helps to overcome the challenges faced by the traditional method [15,20]. PSCM with blockchain offers the following attractive benefits:-

- **End-to-End visibility**: In the blockchain system, every stakeholder can view the changes in the network. This transparency

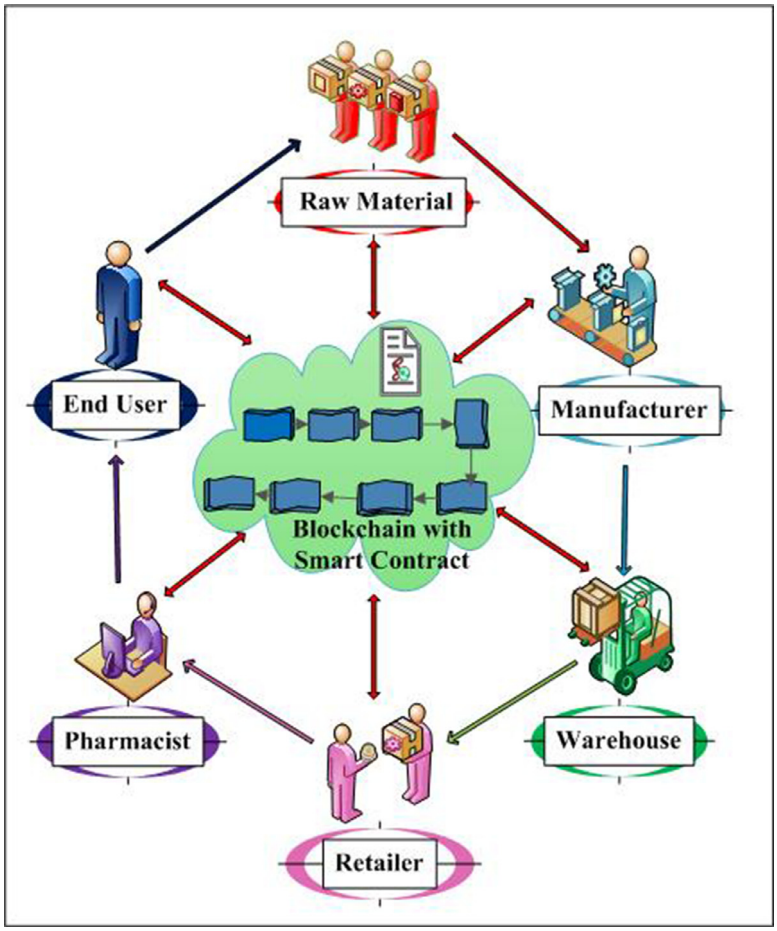

Fig. 2. Overview of pharmaceutical supply chain management system with blockchain.

increases the efficiency of the supply chain, as it is constantly monitored, and identifying the issues will be easier [21].

- **Flexibility**: Failure in a network should create a severe impact. The system should be capable of responding to issues and adapting them as well, without creating any hike on operational cost. It is necessary to sustain in the market.
- **Inferred Trust**: Entities do not need to trust each other, they just need to trust security integrated into the system [22]. Permissioned blockchain-based systems have only authorized users. The system won't allow any transaction which cannot be validated by all the nodes. Smart contracts take care of failures, as well.
- **Control**: The capability of monitoring the channel and rules which govern the system are the key feature of the blockchain system. These features develop a sense of trust for the system.

1.5. Future industry of PSCM with blockchain

Blockchain technology may apply to many problems of the traditional SCM system such as food supply chain, automotive supply chain, seafood verification, pharmaceutical, and drug supply chain, etc. Out of these, the pharmaceutical SCM system is the major concerned for the industry. PSCM is one of the areas which can be benefited by the blockchain technology. In the present scenario, counterfeit drugs are increasing at a rapid rate as the black market provides drugs to innocent people without the knowledge of government. Human life is in danger due to counterfeit drugs. According to the world health organization (WHO) report, the sale of counterfeit drugs is $75 billion in 2010. The developing countries are the major suffering population due to counterfeit drugs. The pharmaceutical companies, as well as a distributor, must improve the traceability and security of the pharmaceutical supply chain system. The blockchain technology has enough potential to resolve the problem related to the PSCM system. Due to the decentralized, distributed, and immutable nature of the blockchain technology, it provides the medicine traceability as well as security of the medicine from the manufacturer to the end customer. The blockchain technology is useful for the pharmaceutical supply chain industry in many ways:-

✓ **Medicine Tracking:-** Large organizations are having lots of elements in its PSCM system. Tracking each, and every item or record is almost impossible for the organization. With the help of a blockchain-based PSC system, tracking of items or records becomes easy, and at the same time, blockchain-based PSCM can detect the fraud in the supply chain system.

✓ **Operational Cost and Time:-** The smart contracts are special scripts in a blockchain environment, which automatically triggered when any specific action takes place [23]. It eliminates the role of intermediaries and middle-man as compared to the traditional supply-chain system. Therefore, operational cost and time are reduced.

✓ **Trust:-** Every industry wants to create a good relationship with their customer. The blockchain is the promising technology that provides the trust to their end customer [23]. The distributed, decentralized and immutable nature of blockchain has enough power to provide the trust-relationship between customer and industry.

1.6. Discussion of existing blockchain-based SCM

Currently, all over the world many companies and startups use the blockchain-based SCM system, to overcome the problems of the traditional supply chain system. Chronicled (san Francisco based) is the blockchain-based startup for the pharmaceutical industry. In 2017, they launched the project to provide security to the supply chain system for gold. After that, they took an initiative called the mediledger for the drug supply chain system. They are expanding their platform for agriculture supply chain, food supply chain, etc. Modum (swiss based start-up) with the University of Zurich, designed a platform for safe delivery of the pharmaceutical medicine [13]. They conducted their first project in 2016. Gemalto [24] with an insurance company launched a project, based on blockchain technology, for the delivery of the medicine from the manufacturers to the hospitals which are located in a very hot climate. They used digital thermometers to record the temperature of the medicine at regular time-interval.

Maersk (Danish shipping company) successfully tested the blockchain to keep track of its shipping containers which is projected for international logistics. To understand the behavior of their project, they tracked a shipment of roses and avocados from Africa to Europe. In the continuation of this, Maersk jointly operated with IBM in September 2016. They applied the proof-of-concept mechanism that tracked a container of flowers from Kenyan to Rotterdam. In 2017-18, Lockheed Martin (defense contracting firm) and the Virginia-based Guard Time Federal launch a project based on the blockchain technology to overcome the supply chain risk (focused on cybersecurity-related initiatives) [24].

Everledger (London-based start-up) adopted blockchain to verify the products provenance (first used for diamonds). After this pilot project, they developed the blockchain-based solution to track wines which are further based on RFID tag. To improve food safety in the SCM system, Walmart and IBM jointly worked on two projects in 2016-17. The first one involves tracking produce from America to the U.S and the second project includes tracking of pork from Chinese farms to Chinese stores. In May 2017, they released the result of the food traceability protocol. According to them, their project reduces the time taken to track the foods (specifically tested on Chinese pork) from days to minutes. Additionally, in 2017, Intel provides a blockchain-based solution for tracking the seafood supply chain.

1.7. Major contributions

To the best of knowledge, integration of blockchain in the supply chain management system is very imperative and necessary. The major contributions of this article are as follows:

✓ We have proposed an architecture for pharmaceutical supply chain management system, and also shown how the information is shared among all involved participants. Moreover, the concept of blockchain technology has also been integrated.

✓ We have also designed, and shown a smart contract mechanism for the same architecture using the state machine model, and further proposed an algorithm for the same.

✓ The consensus algorithm is also one of the important issues in the blockchain-based system. We have also designed a consensus algorithm for the same architecture.

✓ We have proposed a mechanism through which a new entity can join the PSCM network, under the permission of certificate authority, and discussed how the existing entity knows about the new entity. To do this, we develop new key management and its updation protocol through smart contracts.

✓ The proposed smart contract is implemented and evaluated in terms of costs (e.g. transaction cost, execution cost). The computation cost, communication cost, and storage overhead are also discussed in the paper.

✓ Our security analysis shows that the protocol is free from related attacks, and achieves all the requires aspects like confidentiality, integrity, authentication, etc.

1.8. Organization of paper

The rest of the paper is organized as follows: We have introduced our work in Section 1, and then we mentioned the background study of the work such as the overall concept of blockchain, types of blockchain, brief overview of smart contracts mechanism, and consensus algorithm in Section 2. The literature review of this paper is also highlighted in Section 3. Section 4 deals with proposed architecture for PSCM using blockchain mechanism with smart contracts, and consensus algorithm. This section deals with the transaction-message generation, transaction validation, block creation, and block validation protocol. Section 5 deals with key-management in smart contracts, and its updation. Section 6 describes the informal security analysis of the proposed protocol. Section 7 Shows the evaluation of our proposed protocol and smart contract. And finally, in Section 8, we conclude our paper with future scope.

2. Background study

2.1. Basics of blockchain

Blockchain is, in the easiest of terms, an arrangement of unchanging records of information along with time-stamp that is overseen by a group of PCs also known as network entities (or nodes) which are not owned by any single entity [25]. The blocks (set of transactions) in the blockchain network are verified by the majority of nodes. After the verification, the block is added to the chain which is common to all the nodes in a network [26]. Tampering with a single data means that altering the entire chain consisting of thousands of instances that consume a lot of effort, and time, thus making it impossible [27]. The data stored on the blockchain is present as a shared, and continually reconciled database. The decentralization- nature of blockchain reduces the chance of security breach [28]. The main features of the blockchain mechanism are decentralization, transparency, traceability, and immutability, etc.

Public, private, and consortium (or Federated) blockchain are three types of blockchain [29]. In the public blockchain network, the stored data is public i.e visible to everyone within the network. A few examples of public blockchains are ethereum, and bitcoin [30]. In the case of a private blockchain, only the designated users are allowed to get full access or to operate within the network. Hyperledger fabric, corda, and ripple-cryptocurrency are based on private blockchain. Federated Blockchains work under the authority of a certain number of users. Unlike a public blockchain, they don't enable any individual to take part in the verification process. Federated blockchains are faster as compared to the other two, and also are more efficient in terms of privacy, and verification [31].

The smart contract concept was introduced by Nick Szabo in 1994. According to Nick Szabo, smart contracts are "a computerized transaction protocol which executes the term of contracts". Smart contracts are a piece of codes (program or scripts) written in a high-level programming language [32] such as Java, C++, NodeJS, Python, Go, Solidity, etc. Various blockchain platforms use different high-level programming languages for the execution of smart contracts [33]. Hyperledger fabric platform uses NodeJS, Python, Go programming language, on the other side ethereum uses solidity programming language for their smart contracts [34]. A copy

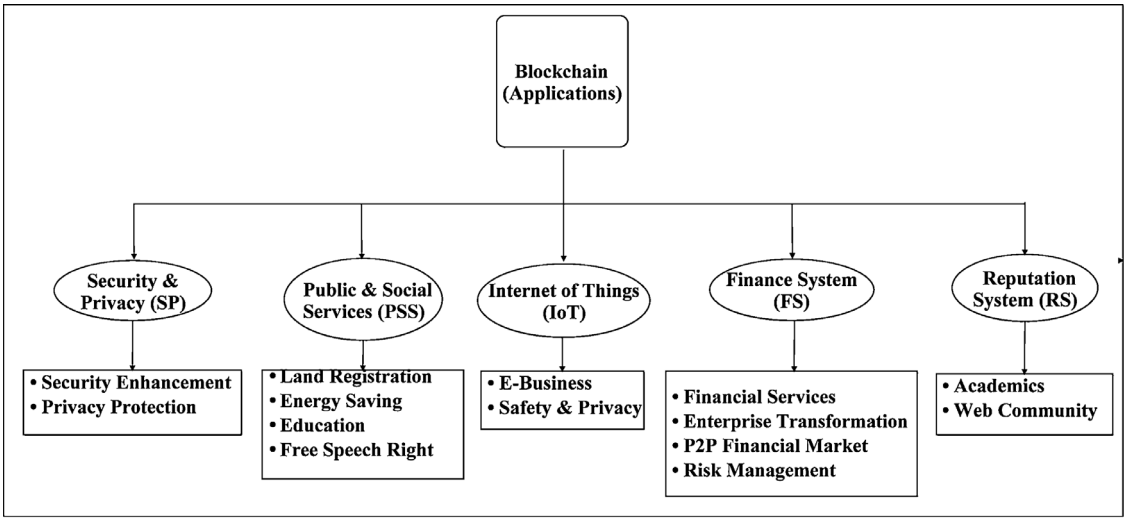

Fig. 3. Applications of blockchain.

of smart contracts is available with every peer in the blockchain-based network. The smart contract scripts are executed automatically, independently, and transparently. It is always executed in a secure environment that provides the correctness of execution, and integrity of code and data [35,36].

The consensus algorithm decides the block-validation process, and it provides consensus among all the nodes, which are available across the decentralized network [37]. For the blockchain-based system, a suitable consensus algorithm is required so that a malicious user can not temper the block, after its execution. Generally, the consensus algorithm is the set of rules to reach a common viewpoint or agreement [38]. The permissionless blockchain-based systems use proof-of-work (PoW), and proof-of-stake (PoS) consensus algorithm, whereas the permissioned blockchain-based systems use byzantine fault tolerance (BFT), and practical byzantine fault tolerance (PBFT) consensus algorithm [39,40].

2.2. Applications of blockchain

The blockchain can be applied in several application areas. Blockchain is capable of storing, history of data in a more efficient manner. Because of its decentralization in nature, transparency, immutability properties, it can be applied in diverse fields [27,41,42]. Some of the important areas where blockchain can be applied are shown in Fig. 3 and the same is explained below:

- **Finance**: The most popular example of blockchain in the field of finance is bitcoin, which was invented by Nakamoto in 2008 [43]. Hyperledger is an open-source blockchain platform started by Linux foundation in 2015 which has brought a tremendous revolution on traditional financial, and business services.
- **Internet-of-Things (IoT)**: Blockchain technology provides a lot of versatile, and decentralized platforms for IoT devices and its applications [31]. IoT gives vast open doors for organizations to execute tasks more brilliantly.
- **Reputation System**: There are numerous limitations to new online reputation platforms. Because of the absence of a severe confirmation mechanism, it fails to gain the client's trust [44]. A decentralized reputation platform would almost certainly fix all the loopholes, and establish trust among clients, and transparency.
- **Public and Social Service**: The primary point of the administration is to keep up a piece of authentic information about people, associations, resources, and activities. A huge amount of cash is spent to keep the record of birth, and demise dates, marital status, business records, property exchanges, or crime, etc. Dealing with this information is exceptionally riotous, and unwieldy. Hence, blockchain gives a simple yet viable approach to oversee, and control every part of information [45].

3. Study of existing works

In the blockchain-based system, blocks are connected with the previous block, using the hash value of previous blocks. Due to this, blockchain provides the immutable blocks, and this is the reason to apply the blockchain mechanism in the PSCM system. Chen et al. [46] proposed the blockchain-based supply chain quality framework. In their framework, they include IoT devices, smart contracts, distributed ledger, and business layers. IoT Devices generate the data for monitoring the quality of the assets. Smart contracts handle the privacy of data and predict the end-user requirements. The business layer includes a business enterprise. Bocek et al. [20] proposed the modum.io, a blockchain-based startup to control the quality of the PSC System. Their framework uses IoT devices to monitor, and measure the temperature of the products. Authors utilize the ethereum based platform for the execution of smart contracts. The smart contracts are written in a solidity programming language.

Kim et al. [47] applied ontologies for the supply chain using the blockchain mechanism. The required ontology axioms represented in the form of first-order logic, and later it is converted into the ethereum based smart contracts (solidity based) to provide the traces of goods. Figurili et al. [48] applied Azure blockchain workbench to trace the woods from raw material to the final product. In their work, they utilize the RFID sensor devices to capture the data. Mitsuaki [49] proposed a blockchain-based solution to solve the information asymmetry and double marginalization problems of the supply chain management system. Their work utilizes the homomorphic encryption method to provide the security of the user data with blockchain. Smart contracts mechanism and other legal issues were not discussed by the authors. Tian [50] used RFID along with a blockchain mechanism to provide a traceability system for an agri-food supply chain in China. The author claims that their system provides information tracing, guarantee freshness of products, and transparency of product information. To achieve both anonymity and regulation properties, Lin et al. [51] proposed a conditional anonymous pay-

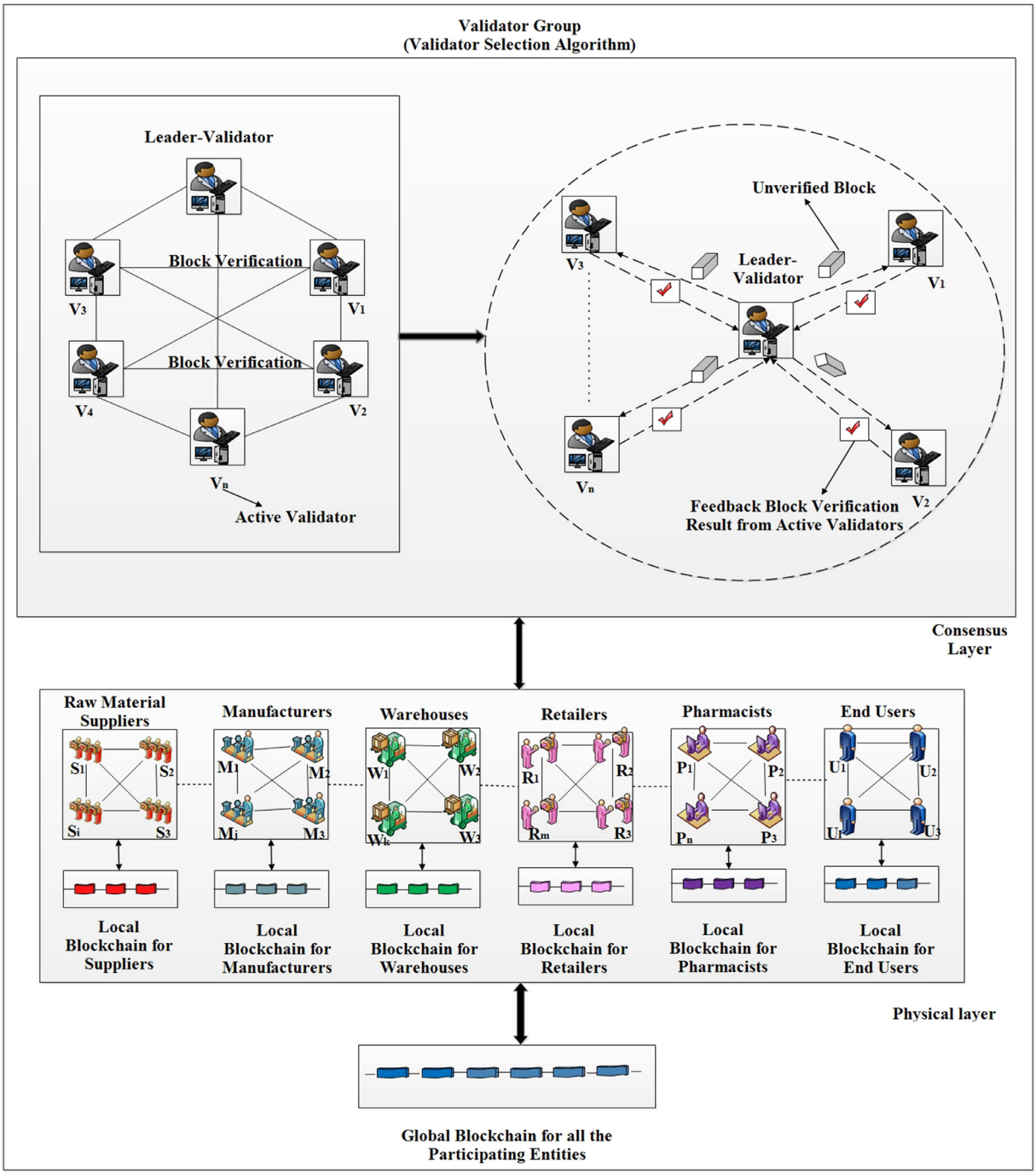

Fig. 4. Proposed architecture.

ment scheme and used it constructed the first decentralized conditional anonymous payment system. Comparisons with that of Zerocash show the proposed system is practical for real-world deployment.

4. Proposed architecture for PSCM

4.1. High-level description

As shown in Fig. 4, the proposed blockchain-based PSCM system consist of suppliers (S_i), manufacturers (M_j), warehouses (W_k), retailers (R_m), pharmacists (P_n), and end-users (U_l). The various notations used in the proposed blockchain-based PSCM system are illustrated in Table 1.

The entire PSCM is divided into multiple groups according to its services. For example, all the suppliers are in the same group. The entities within the same group create a decentralized peer-to-peer network. In each group, two types of nodes are present: normal nodes and validator nodes. The validator nodes have the high computing power than normal nodes that are responsible for the generation of transaction message $\langle T_{r_i} \rangle$ as well as validation of transaction and block $\langle T_{r_i}, B_i \rangle$. The normal nodes have less computational power, and they are not participating in the validation process. These set of validator nodes create a separate validator group for the validation of transaction and a block $\langle T_{r_i}, B_i \rangle$. The proposed

Table 1
Descriptions of various notations used in proposed scheme.

Notations	Description
(S_i)	Supplier (S_i)
(M_j)	Manufacturer (M_j)
(W_k)	Warehouse (W_k)
(R_m)	Retailer (R_m)
(P_n)	Pharmacist (P_n)
(U_l)	End user (U_l)
L_v	Leader-validator node L_v
V_x	Remaining validators V_x where $(x = 1, 2, 3 \ldots n)$
(W_z)	Warehouse (W_z)
(T_{r_i})	Transaction (T_{r_i})
T_{id_i}	Transaction id T_{id_i} corresponding to (T_{r_i})
ID_{S_i}	Identity of (S_i)
ID_{W_z}	Identity of (W_z)
$P_{J_{Lv}}$	Public-key of L_v
$P_{R_{Lv}}$	Private-key of L_v
$P_{U_{Wz}}$	Public-key of W_z
$P_{R_{Wz}}$	Private-key of W_z
$P_{U_{CA}}$	Public-key of (CA)
$P_{T_{CA}}$	Private-key of (CA)
B_i	Block Number (i)
$H(\cdot)$	Cryptographic hash function
$E(\cdot)$	Encryption function
$D(\cdot)$	Decryption function
$\|\|$	Concatenation operator

system comprises a global blockchain and a local blockchain for each group. The global blockchain is maintained by all the entities in the system, whereas each local blockchain is maintained by entities of a particular group.

The proposed blockchain scheme for a secure information sharing system for PSCM is shown in Fig. 5. The system comprises the following phases: transaction message generation, transaction message validation, block creation, and block validation. The detail description of all the these phases are given in the Section 4.2–4.5.

All the entities of the network are downloaded and updated with the most recent copy of the blockchain. In our scheme, the blockchain acts as a distributed public ledger, which stores transaction of all the entities of the network. Whenever any entity of the network performs a transaction $\langle T_{r_i} \rangle$ with any other entity, the initiating entity stores the transaction $\langle T_{r_i} \rangle$, and forwards the same transaction to the leader-validator node $\langle L_v \rangle$ [52] for transaction-validation. If transaction-validation is successful, the leader-validator node creates a new block $\langle B_i \rangle$. The remaining validator nodes $\langle V_x \rangle$ validate this new block $\langle B_i \rangle$. If block-validation is successful, then only the leader-validator adds the new block into the blockchain network. All other entities of the network, synchronize with the leader-validator node, to get the most updated copy of the blockchain.

4.2. Transaction message generation

The system considers both local, as well as global transactions. Local transactions (for example: supplier (S_1) to supplier (S_2)) are stored in the local blockchain, while global transactions (for example: supplier (S_i) to warehouse (W_j)) are stored in the global blockchain. Whenever one entity performs the transaction with any other entity, the same transaction also sends to the leader-validator node for the transaction-validation. The leader-validator node first verifies the transaction based on the signature of initiator-node, and with some other parameters such as transaction-ID, the identity of initiator-node. If the transaction is verified correctly, the leader-validator node creates a new block, otherwise discards the transaction with an error message: "transaction was not verified correctly".

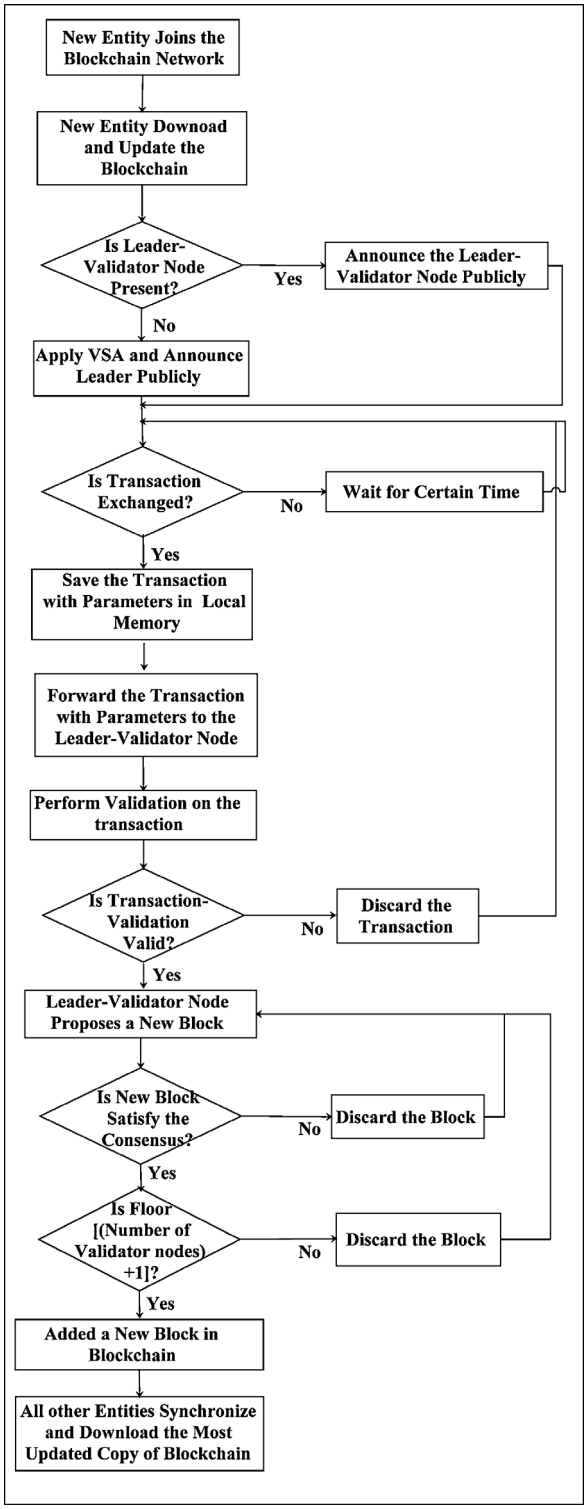

Fig. 5. Flow chart model of the proposed PSCM.

4.3. Transaction message validation

We assume that, the transaction T_{r_i} was exchanged from the supplier (S_1) to supplier (S_2). To validate the transaction T_{r_i}, the Validator Selection Algorithm (VSA) [52] selects the leader-validator node L_v among all the validator nodes V_x, and disseminates the same in the local group of suppliers S_i. The above process ensures that every entity in the group knows the leader-validator

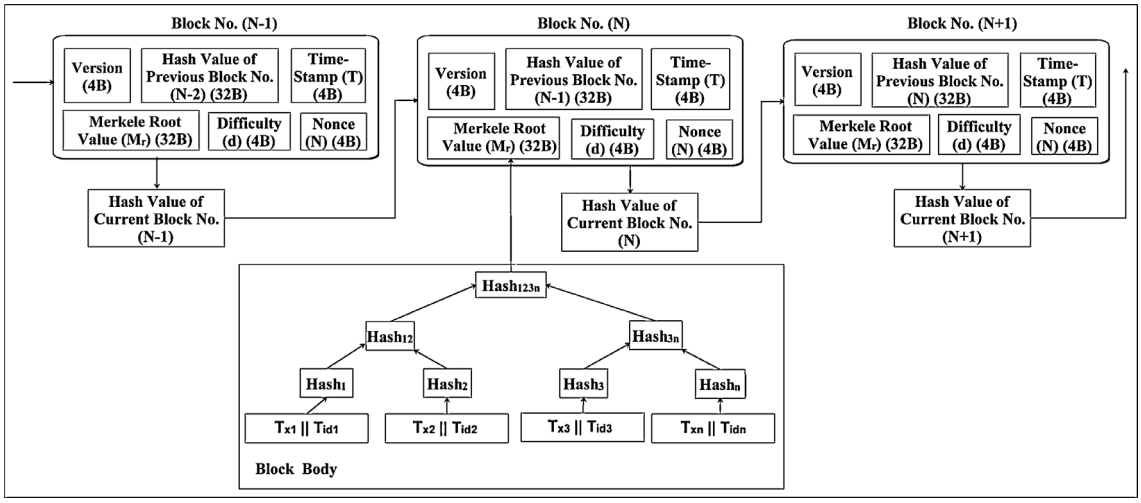

Fig. 6. Structure of block in blockchain.

node L_v. The initiator node S_1 sends the same transaction T_{r_i} to the leader-validator node L_v for transaction-validation.The following sequence of steps are carried out, to validate the transaction T_{r_i}.

Step 1. S_1 performs the $H(.)$ on T_{r_i}, T_{id_i}, ID_{S_1}, and computes the parameter $X = H(T_{r_i} \parallel T_{id_i} \parallel ID_{S_1})$.

Step 2. Now, S_1 performs digital signature on T_{r_i}, and applies the encrypt operation $E(.)$ on it by using public key $P_{U_{L_v}}$ of leader-validator node to get the parameter Y.

Step 3. S_1 sends the parameters $\langle X, Y, T_{id_i}, ID_{S_1} \rangle$ to L_v publicly.

Step 4. On receiving the parameters $\langle X, Y, T_{id_i}, ID_{S_1} \rangle$, L_v applies the decrypt operation $D(.)$ on $\langle Y \rangle$. After decrypting, due to the signature of S_1 on the T_{r_i}, L_v validates that S_1 initiates T_{r_i}.

Step 5. L_v computes $X' = H(T_{r_i} \parallel T_{id_i} \parallel ID_{S_1})$, and checks whether received parameter $\langle X \rangle$ is same as calculated parameter $\langle X' \rangle$ or not?

Step 6. If $X' = X$ is correct, L_v creates a new block $\langle B_i \rangle$ based on the T_{r_i} and T_{id_i}, Otherwise discards the transaction with error message: "transaction validation is not successful".

Proof for the condition ($X' = X$): In our proposed protocol, S_1 forwards the $\langle X, Y, T_{id_i}, ID_{S_1} \rangle$ parameters to L_v. To do this, S_1 first uses the standard digital signature method on $\langle T_{r_i} \rangle$ and then applies the $E(.)$ on it by using the $P_{U_{L_v}}$ to get the parameter $\langle Y \rangle$. The parameter $\langle X \rangle$ depends on the $\langle T_{r_i}, T_{id_i}, ID_{S_1} \rangle$ and $H(.)$ operation of $\langle T_{r_i}, T_{id_i}, ID_{S_1} \rangle$ provides $\langle X \rangle$. In order to provide the correctness of the condition, we assume that an attacker changes the few parameters from $\langle X, Y, T_{id_i}, ID_{S_1} \rangle$ and the updated value is $\langle X, Y, T'_{id_i}, ID'_{S_1} \rangle$. On receiving the parameters $\langle X, Y, T'_{id_i}, ID'_{S_1} \rangle$, L_v performs the $D(.)$ operation on $\langle Y \rangle$ by using the $P_{R_{L_v}}$ to get $\langle T_{r_i} \rangle$. After getting the $\langle T_{r_i} \rangle$, L_v performs the $H(.)$ operation on $\langle T_{r_i}, T'_{id_i}, ID'_{S_1} \rangle$ and gets value (X'). As a result the equality condition ($X' = X$) does not hold and L_v knows that $\langle T_{r_i}, T_{id_i} \rangle$ is not a valid transaction.

4.4. Block creation

After the transaction-validation procedure, the leader-validator node (L_v) creates a new block $\langle B_i \rangle$. The block consists of a block header, and a block body. Block header contains the previous block hash (P_h), merkle root (M_r), nonce (N), time-stamp (T_i) for block creation, block version (B_V), a random difficult number (d) for desired target value, and block body contains the transaction (T_{r_i}) with transaction-ID (T_{id_i}) as shown in Fig. 6. The merkle tree which is binary tree, is created by using the transaction (T_{r_i}), and transaction-ID (T_{id_i}), and root value of the merkle tree is stored on the block header for further computation. The (L_v) computes the current block hash (C_h) for a new block $\langle B_i \rangle$ based on the previous block hash (P_h), merkle root (M_r), nonce (N), and time-stamp (T_i). The mathematical formula to achieve this is:

$$(C_h) = H((P_h) \parallel (M_r) \parallel (N) \parallel (T_i)) \tag{1}$$

Initially, nonce value sets to zero, and at each iteration, it is incremented by one. The block values (P_h), (M_r), (T_i) are repeatedly hashed with different values of (N) to create a suitable current hash index (C_h), which satisfies the difficulty-target (d). The structure of both local, and global blockchains is the same. A transaction message is stored in the local blockchain if and only if, a transaction occurs between the entities of the same group, otherwise the transaction is stored in the global blockchain network.

4.5. Block validation

Once the new block $\langle B_i \rangle$ has created by the leader-validator node (L_v), the (L_v) sends encrypted block $\langle B_i \rangle$ to the remaining validator nodes for the block-validation. The other validator nodes verify $\langle B_i \rangle$ and respond with the acknowledgment message, either 0 or 1 to the leader-validator node. We assume that 0 stands for negative acknowledgment i.e. $\langle B_i \rangle$ is not verified correctly by the other validator nodes, whereas 1 stands for positive acknowledgment i.e. $\langle B_i \rangle$ is verified correctly by the other validator nodes. If more number of validator nodes verify $\langle B_i \rangle$ correctly, then $\langle B_i \rangle$ is accepted by (L_v), which is shown in Fig. 4. After validation of $\langle B_i \rangle$, (L_v) adds $\langle B_i \rangle$ in the blockchain network, either in a local or global blockchain network, based on the local or global transaction. All the remaining entities of the network synchronize with the (L_v) to get the most updated copy of the blockchain.

The following sequence of steps is carried out, to validate the new block $\langle B_i \rangle$.

Step 1. (L_v) performs the $E(.)$ operation on $\langle C_h, M_r, d, T_i \rangle$ using $P_{R_{L_v}}$, and sends to the other remaining validator nodes (V_x).

Step 2. The other validator nodes (V_x) (where $x = 1, 2, 3 \ldots n$) perform the $D(.)$ operation on $\langle C_h, M_r, d, T_i \rangle$ using $P_{U_{L_v}}$.

Step 3. After getting $\langle M_r, d, T_i \rangle$ parameters, all the (V_x) compute $(C_{h'}) = H((P_h) \parallel (M_r) \parallel (N) \parallel (T_i))$. The value of (N) is chosen such that it satisfies (d).

Step 4. If $(C'_h) = (C_h)$ is correct, all the (V_x) respond with acknowledgement (either 0 or 1) to (L_v).

Step 5. If $(\lfloor (N/2) \rfloor + 1)$ number of (V_x) respond with the trustworthy acknowledgement message to (L_v), then (L_v) accepts $\langle B_i \rangle$, and adds the $\langle B_i \rangle$ in the blockchain network.

Proof for the condition $(C'_h = C_h)$: In our proposed protocol, L_v forwards the $\langle C_h, M_r, d, T_i \rangle$ parameters to all the remaining (V_x) which exist in the network. To do this, L_v applies the $E(.)$ operation on $\langle C_h, M_r, d, T_i \rangle$ by using the $P_{R_{L_v}}$. In order to provide the correctness of the condition, we assume that an attacker changes the few parameters from $\langle C_h, M_r, d, T_i \rangle$ and the updated values are $\langle C_h, M_r, d', T'_i \rangle$. After getting $\langle C_h, M_r, d', T'_i \rangle$ parameters, the remaining (V_x) perform the $H(.)$ operation on $\langle C_h, M_r, N', T'_i \rangle$. The value of N depends on d. Since d is changed by the attacker, correspondingly N is also changed and we assume that new value is (N'). As a result, equality condition $(C'_h = C_h)$ does not hold and all the (V_x) know that $\langle B_i \rangle$ is not a valid block.

4.6. Smart contract mechanism for proposed blockchain-based PSCM

Smart contracts are the programs (scripts) that are written in the high-level programming language. These scripts are stored in every node of the blockchain-based network. If any node wants to perform a transaction with any other node, smart contracts check whether the transaction is according to predefined rules (scripts) or not. If it is according to rules (scripts), then the node can execute his/her transaction otherwise, the system throws an error message, stating that "transactions cannot be completed". Generally, smart contracts are based on the state-machine model, which follows the deterministic approach. The deterministic state machine model is represented by a directed graph, where vertices (node) represents the state of the machine, and an edge represents the transitions from one state to another state. The smart contract algorithm for proposed Blockchain-based PSCM is given as Algorithm 1.

Our proposed smart contracts which are based on the state machine model consists of 6 states: labeled as state 0, state 1, state 2, state 3, state 4, and state 5. These states represent different entities. state 0 represents the manufacturer entity, state 1 represents the warehouse entity, state 2 represents the retailer entity, state 3 represents the pharmacist entity, state 4 represents the end-user entity, and state 5 represents the dead state. In our proposed model, various actions are defined. These actions are used as transitions from one state to another state. Set of actions are Purchase, Delivery, Demand, Supply, No action, and Violation. For example, if the machine is in state 1, and action is delivery then the machine automatically moves from state 1 to state 2. If the machine is in state 3, and action is a violation then the machine moves from state 3 to state 5 with an error message. If the machine is in state 4, and action is demand then the system moves from state 4 to state 3, and so on. According to the state of the machine, ownership of the asset will be changed. The proposed state machine model is shown in Fig. 7.

Algorithm 1 Smart contract algorithm for proposed blockchain-based PSCM.

Input: *Actions such as Purchase, Delivery, Demand, Supply, No Action, Violation*

Output: *Messages such as Ownership of Product, Error and abort, No Action required*

```
if (f(Action) == Purchase) then
    if (Product is available) then
        fun(Action) = Delivery;
    else
        Report with message, product is not available
    end if
end if

if (f(Action) == Delivery) then
    if (Delivery_Date < Expiry_Date) then
        if (Delivery_Date < Predefined_Time_Delivery) then
            Provide the product to other entity and change the
            ownership of the product;
        else
            Report with message, product is not delivered;
        end if
    else
        Report with error message, product was expired;
    end if
end if

if (f(Action) == Demand) then
    if (Product is available) then
        if (Delivery_Date < Expiry_Date) then
            fun(Action) =Supply;
        else
            Report with message, product was expired;
        end if
    else
        Report with message, product is not available;
    end if
end if

if (f(Action) == Supply) then
    if (Product is available) then
        Deliver the product to end user and change the ownership
        of the product;
    else
        Report with message, product is not available;
    end if
end if

if (f(Action) == No Action) then
    Print: Entity will remain in the same state;
end if

if (f(Action) == Violation) then
    Print: Report with error message showing that wrong operations has been performed;
end if
```

4.7. Consensus mechanism for proposed blockchain-based PSCM

The consensus mechanism is the set of rules to provide, a common agreement among nodes. Generally, the consensus algorithm is designed in such a way that after executing the new block, the majority of nodes in the network agree that new block is a valid block, and can be included in the blockchain network. Once, they reach into the consensus, the nodes cannot change their decision. For the proposed blockchain based PSCM, we designed a consensus mechanism for the validation of transaction and new block. The validation of the transaction is done by the leader-validator node as per discussion in Section 4.3. Once the validation procedure is successful, the leader-validator node creates a new block. The remaining validator nodes perform the block-validation procedure as per discussion in Section 4.5. The consensus mechanism is given in Algorithm 2.

Algorithm 2 Consensus algorithm for proposed blockchain-based PSCM.

Input: New transaction = T_{r_i}, *BlockChain Length* = $BC[l]$, $BC[l] = B_0$, Difficulty_Target

Output: *new block* = $\langle B_i \rangle$, *Current hash of new block* = $\langle C_h \rangle$, *Current Blockchain length* $BC[l]$, *Nonce*

Perform transaction T_{r_i};
Check_Validation of T_{r_i};
if (Validation==TRUE) **then**
    Print: Validation of T_{r_i} is successful;
    Leader-validator node (L_v) creates a new block $\langle B_i \rangle$;
else
    Print: Generate an error message: "validation is not successful";
end if
Compute $\langle C_h \rangle$ of $\langle B_i \rangle$;
Check_Validation of $\langle B_i \rangle$;
if (Validation==TRUE) **then**
    Print: Validation of $\langle B_i \rangle$ is successful;
else
    Print: Generate an error message: "validation is not successful";
end if
while (TRUE) **do**
    if ($\langle C_h \rangle$ == *Difficulty_Target*) **then**
        Add BC(new_block);
        $BC[l] = BC[l] + 1$;
    else
        Change(Nonce);
    end if
end while

5. Key-Management in smart contracts

The proposed blockchain-based PSCM system consists of many entities such as suppliers (S_i), manufacturers (M_j), warehouses (W_k), retailers (R_m), pharmacists (P_n), and end users (U_l). If any new entity (W_z) wants to join the PSCM, (W_z) requests to (CA) for issuing the certificate. To do this, (CA) first verifies the (W_z) based on the encrypted parameters, which (W_z) sent to (CA). After the verification procedure, (CA) checks $\langle P_{U_{W_z}} \rangle$ corresponding to $\langle ID_{W_z} \rangle$ in its database as shown in Fig. 8. If $\langle P_{U_{W_z}} \rangle$ corresponding to $\langle ID_{W_z} \rangle$ is not exist, then only (CA) issue the certificate (CER_{W_z}) to (W_z). The certificate contains the name of (W_z), W_z public key $\langle P_{U_{W_z}} \rangle$, W_z identity $\langle ID_{W_z} \rangle$, certificate validation date, version number, (CA) name, and (CA) digital signature. Once (W_z) gets (CER_{W_z}), (W_z) forwards the (CER_{W_z}) to (L_v) publicly for certificate verification, whether (CA) is generated (CER_{W_z}) or not? This certificate verification ensured that (CA) is generated (CER_{W_z}) to (W_z).

If verification of (CER_{W_z}) is correct, then (L_v) believes that $\langle P_{U_{W_z}} \rangle$ is correct, and it belongs to (W_z). After this, (L_v) updates the smart contracts with public key $\langle P_{U_{W_z}} \rangle$ corresponding to (W_z) identity $\langle ID_{W_z} \rangle$. The remaining (V_x), and entities synchronize with the (L_v) to get the most updated copy of smart contracts. The tasks involved in updating the smart contracts module with new block creation for PSCM system are shown in Fig. 8. The following sequence of steps are carried out, if (W_z) wants to join the PSCM network:

Step 1. (W_z) chooses $\langle ID_{W_z} \rangle$, and generates a pair of keys $\langle P_{U_{W_z}}, P_{R_{W_z}} \rangle$ using public key generator (PKG). (W_z) keeps $\langle P_{R_{W_z}} \rangle$.

Step 2. (W_z) performs the $H(.)$ operation on (ID_{W_z}, $P_{U_{W_z}}$, T_s) and computes the parameter $A = H(ID_{W_z} \parallel P_{U_{W_z}} \parallel T_s)$, where ($T_s$) is the time when pair of keys are generated.

Step 3. (W_z) performs the $E(.)$ operation on $\langle ID_{W_z}, P_{U_{W_z}}, T_s, A \rangle$ using $P_{U_{CA}}$, and sends to the (CA) publicly.

Step 4. On receiving the encrypted parameters, (CA) applies $D(.)$ operation on $\langle ID_{W_z}, P_{U_{W_z}}, T_s, A \rangle$ using $P_{R_{CA}}$.

Step 5. (CA) computes $A' = H(ID_{W_z} \parallel P_{U_{W_z}} \parallel T_s)$, and checks whether received parameter $\langle A \rangle$ is same as calculated parameter $\langle A' \rangle$ or not?

Step 6. If $A' = A$ is correct, then (CA) recognizes that the $\langle P_{U_{W_z}} \rangle$ belongs to (W_z), and issue the (CER_{W_z}) to (W_z).

Step 7. (W_z) forwards the (CER_{W_z}) to (L_v) for certificate verification.

Step 8. (L_v) performs the $D(.)$ operation on (CA) digital signature using $P_{U_{CA}}$ to get the original message digest.

Step 9. After the above procedure, (L_v) performs the $H(.)$ operation on the name of (W_z), $\langle P_{U_{W_z}} \rangle$, $\langle ID_{W_z} \rangle$, certificate validation date, version number, (CA) name, parameters to get the another message digest.

Step 10. If the original message digest is same as calculated one, then (L_v) recognizes that (CA) is the issuer of (CER_{W_z}).

Step 11. After the verification procedure, (L_v) adds $P_{U_{W_z}}$ in the smart contracts module.

Proof for the condition ($A' = A$): In our proposed protocol, (W_z) forwards the $\langle ID_{W_z}, P_{U_{W_z}}, T_s, A \rangle$ parameters to (CA). To do this, (W_z) applies the $E(.)$ operation on $\langle ID_{W_z}, P_{U_{W_z}}, T_s, A \rangle$ by using $P_{U_{CA}}$. The parameter (A) depends on the $\langle ID_{W_z}, P_{U_{W_z}}, T_s \rangle$ and $H(.)$ operation of $\langle ID_{W_z}, P_{U_{W_z}}, T_s \rangle$ provides (A). In order to provide the correctness of the condition, we assume that an attacker changes the few parameters from $\langle ID_{W_z}, P_{U_{W_z}}, T_s, A \rangle$ and the updated values are $\langle ID'_{W_z}, P'_{U_{W_z}}, T'_s, A \rangle$. After getting the parameters $\langle ID'_{W_z}, P'_{U_{W_z}}, T'_s, A \rangle$, ($CA$) performs the $H(.)$ operation on $\langle ID'_{W_z}, P'_{U_{W_z}}, T'_s \rangle$ and gets the value (A'). As a result, the equality condition ($A' = A$) does not hold and (CA) does not issue the (CER_{W_z}) to (W_z).

We assume that, the generation of $\langle P_{U_{W_z}} \rangle$ by (W_z) is also one transaction. The same transaction is first verified by the (L_v) as per discussion in Section 4.3. After the transaction-verification procedure, (L_v) creates a new block $\langle B_j \rangle$ as per discussion in Section 4.4. Once (L_v) creates $\langle B_j \rangle$, remaining (V_x) verify the $\langle B_j \rangle$ as per discussion in Section 4.5. If the block-validation is successful, then (L_v) adds the $\langle B_j \rangle$ in global blockchain network. All the remaining entities know that new entity (W_z) joins the network, and they update their smart contracts module. The benefit of this scheme is that every participating entity of the network knows about remaining entities and their public keys which is beneficial for future transactions.

6. Security analysis of our proposed protocol

In this section, we analyze whether an adversary A can launch attacks or not, based on our protocol.

In our protocol, S_1 forwards $\langle X, Y, T_{id}, ID_{S_1} \rangle$ to L_v publicly. We assume that adversary A gets $\langle X, Y, T_{id}, ID_{S_1} \rangle$ from public network, where $X = H(T_{r_i} \parallel T_{id} \parallel ID_{S_1})$, and $E(.)$ operation on (T_{id}) using $P_{U_{L_v}}$ provides the parameter Y. There are few cases arise.

- **Case 1:** Adversary A changes (T_{id}) to (T'_{id}). L_v decrypts the parameter $\langle Y \rangle$ using $P_{R_{L_v}}$, and computes the parameter $(X') = H(T_{r_i} \parallel T'_{id} \parallel ID_{S_1})$. As a result $(X' \; != X)$, and L_v predicts that some adversary **A** changes either (T_{id}) or (ID_{S_1}). So even though the adversary **A** gets parameters $\langle X, Y, T_{id}, ID_{S_1} \rangle$ from the public network, he/she cannot break the security of the proposed system.

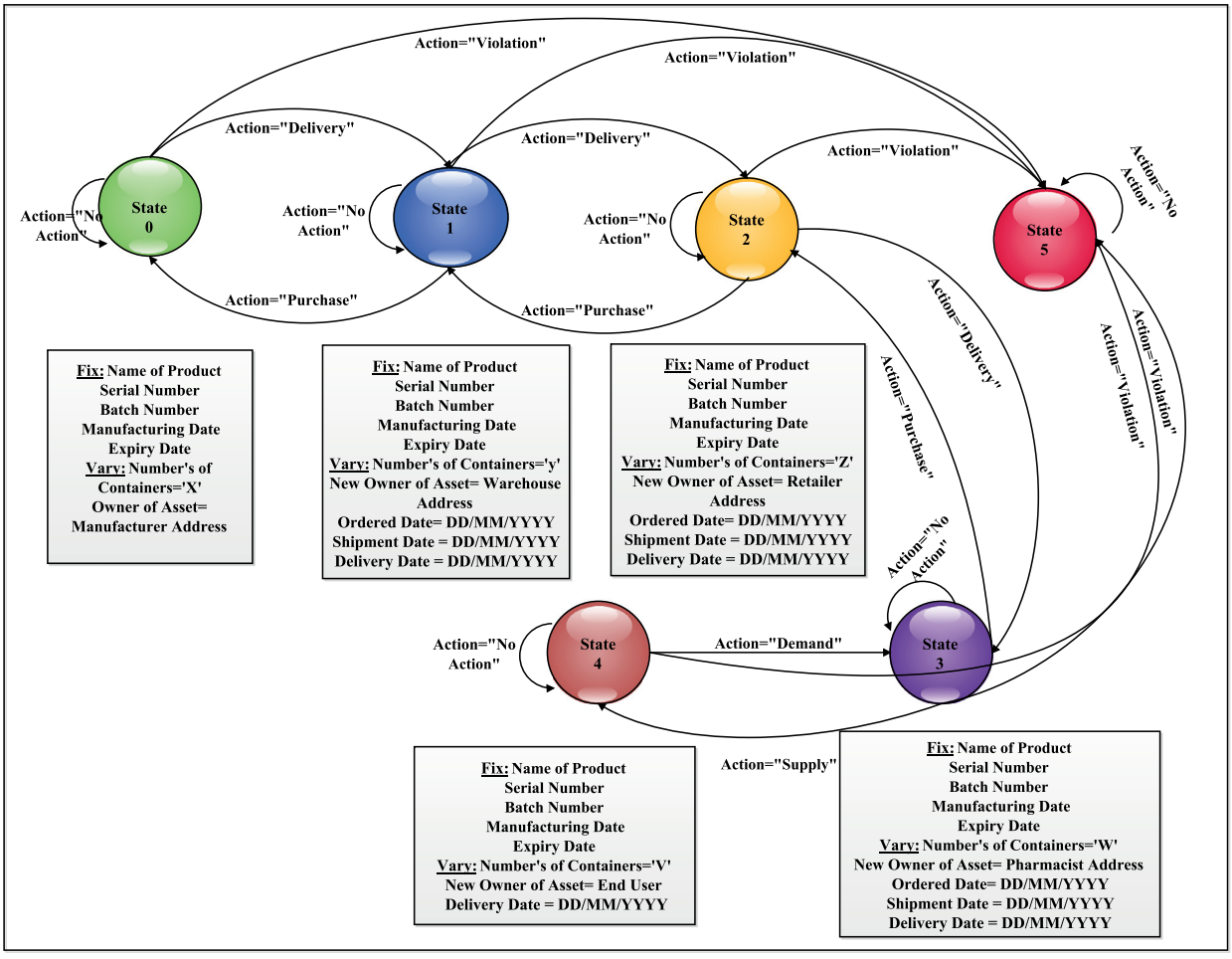

Fig. 7. Smart contracts as state machine for proposed blockchain-based PSCM.

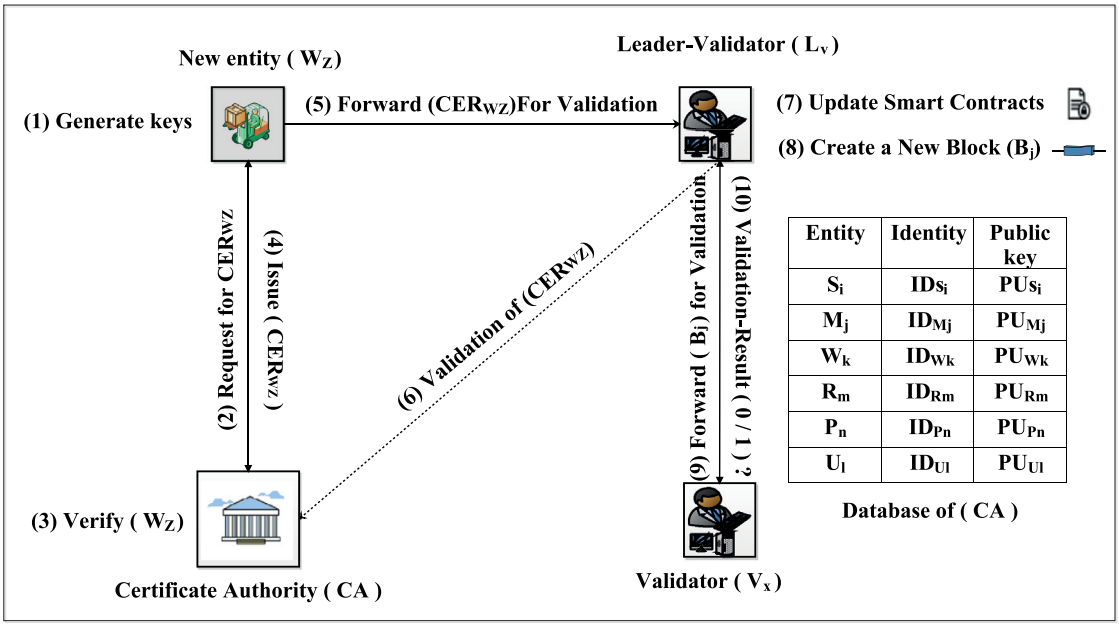

Fig. 8. Key-management and smart contracts updation.

- **Case 2:** Adversary A changes (ID_{S_1}) to (ID'_{S_1}). L_v decrypts the parameter $\langle Y \rangle$ using $P_{R_{L_v}}$ and computes the parameter $(X') = H(T_{r_i} \parallel T_{id} \parallel ID'_{S_1})$. As a result $(X' \; != X)$, and L_v predicts that some adversary **A** changes either (T_{id}) or (ID_{S_1}). So even though the adversary **A** gets parameters $\langle X, Y, T_{id}, ID_{S_1} \rangle$ from the public network, he/she cannot break the security of the proposed system.
- **Case 3:** Adversary A changes (T_{id}) to (T'_{id}) and (ID_{S_1}) to (ID'_{S_1}). L_v decrypts the parameter $\langle Y \rangle$ using $P_{R_{L_v}}$ and computes the parameter $(X') = H(T_{r_i} \parallel T'_{id} \parallel ID'_{S_1})$. As a result $(X' \; != X)$, and L_v predicts that some adversary **A** changes parameter $\langle (T_{id}), (ID_{S_1}) \rangle$. So even though the adversary **A** gets parameters $\langle X, Y, T_{id}, ID_{S_1} \rangle$ from the public network, he/she cannot break the security of the proposed system.

In our protocol, L_v performs the $E(.)$ operation on $\langle C_h, M_r, d, T_i \rangle$ using $P_{R_{L_v}}$, and sends to (V_x) (where $x = 1, 2, 3 \ldots n$) publicly for block-validation. We assume that adversary **A** is able to decrypts it, and changes few parameters. There are few cases arise.

- **Case 1:** Adversary A changes (d) to (d'). As a result, value of (N) also changes. We assume that the new value of (N) is (N'). (V_x) computes $(C_{h'}) = H((P_h) \parallel (M_r) \parallel (N') \parallel (T_i))$. As a result, (C'_h) $!= (C_h)$, and (V_x) predicts that some adversary **A** changes either (d) or (T_i). So even though the adversary **A** gets parameters $\langle C_h, M_r, d, T_i \rangle$ from the public network, he/she cannot break the security of the proposed system.
- **Case 2:** Adversary A changes (T_i) to (T'_i). (V_x) computes $(C_{h'}) = H((P_h) \parallel (M_r) \parallel (N) \parallel (T'_i))$. As a result, (C'_h) $!= (C_h)$ and (V_x) predicts that some adversary **A** changes either (d) or (T_i). So even though the adversary **A** gets parameters $\langle C_h, M_r, d, T_i \rangle$ from the public network, he/she cannot break the security of the proposed system.
- **Case 3:** Adversary A changes (d) to (d') and (T_i) to (T'_i). (V_x) computes $(C_{h'}) = H((P_h) \parallel (M_r) \parallel (N') \parallel (T'_i))$. As a result, (C'_h) $!= (C_h)$ and (V_x) predicts that some adversary **A** changes $\langle (d), (T_i) \rangle$. So even though the adversary **A** gets parameters $\langle C_h, M_r, d, T_i \rangle$ from the public network, he/she cannot break the security of the proposed system.

7. Performance evaluation

7.1. Calculation of computation and communication cost

This section presents the performance evaluation of the proposed blockchain-based PSCM scheme in terms of security properties computation cost and communication cost. These parameters are the most important factors to measure the performance of the authentication protocol.

To calculate the computation cost of Proposed Protocol, we consider one entity supplier (S_1), leader-validator node (L_v), one more validator (other than leader-validator) (V_x), and one new entity warehouse (W_z). This paper mainly uses the light weight encryption $E(.)$, and decryption $D(.)$ functions to compute T_e and T_d respectively. 256-bit one way cryptographic hash function $H(.)$ is used to compute T_h. (PKG) is used to generate the key-pair of (S_1) and (W_z). In Table 2, we have summarized the computation cost of the proposed protocol respectively.

T_{pg}: Execution time for PKG function
T_e: Execution time for encryption $E(.)$
T_d: Execution time for decryption $D(.)$
T_h: Execution time for One-way hash function $H(.)$

It can reasonably be assumed that the length of $\langle T_{id}, ID_{S_1}, ID_{W_z}, P_{U_{W_z}} \rangle$, and parameter $\langle Y \rangle$, each takes 128 bits. Message digest $H(\cdot)$ and parameters $\langle X, A \rangle$, each takes 256 bits. whereas parameters $\langle d, T_i, T_s \rangle$, each takes 32 bits for measuring the communication cost of the proposed protocol. In Table 3, we have summarized the communication cost of the proposed protocol.

Table 2
Computation cost of proposed protocol.

Entity	Cost
(S_1)	$1T_{pg} + 1T_h + 1T_e$
(W_z)	$1T_{pg} + 1T_h + 1T_e$
(L_v)	$4T_h + 1T_e + 2T_d$
(V_x)	$1T_h + 1T_d$
(CA)	$1T_h + 1T_d$
Total	$2T_{pg} + 8T_h + 3T_e + 4T_d$

Table 3
Communication cost of proposed protocol.

Communication Mode	Cost (bits)
$(S_1) \rightarrow (L_v)$	$256 + (128 * 3)$
$(L_v) \rightarrow (V_x)$	$(256 * 2) + (32 * 2)$
$(W_z) \rightarrow (CA)$	$256 + (128 * 2) + 32$
Total	$(256 * 4) + (128 * 5) + (32 * 3)$

7.2. Storage overhead of the proposed scheme

As shown in Fig. 6, the size of the block header for the proposed blockchain-based PSCM scheme is about 80*Bytes*, and the size of each transaction is 4096*Bytes* approximately. Therefore, the total size of one block with a single transaction is 4176*Bytes*. To prevent attacks, we also assume that in 120 sec.120one block is generated. Therefore, the size of blockchain with a single transaction is 2.867*MB* per day. We carefully investigated the structure of the block for the PSCM network, which can support the huge amount of data. We assume different cases, where one participating entity exchanges information with others for a given time- period.

The general mathematical formula to calculate the size of blockchain is:

$$Size = T_x * B_s * T \tag{2}$$

where (T_x), (B_s), (T) denotes the number of transactions for a time period, the size of block, and time (in units) respectively [53].

If the participating entities of the PSCM network generate 100 number of transactions per second and each transaction is the size of 4176*Byte* then according to the Eq. 2, the size of blockchain with 100 number of transactions for per minute is calculated as $= (4176*100*60)/(1024*1024) = 23.89MB$ per minute. Here, $(T_x) = 100$, $(B_s) = 4176B$, $(T) = 60sec$. We consider the different cases in which the number of transactions is continuously increasing and calculated what is the size of blockchain for a different time period that is shown in Table 4.

7.3. Smart contract evaluation

The proposed smart contracts for the blockchain-based PSCM system is evaluated by writing the programs in a solidity programming language. Solidity version 0.5.10 and REMIX IDE with a specification of Intel (R) Core (TM) $i5 - 8250U$ CPU @1.60*GHz*, 8*GB* of RAM, *Win*10, $64 - bit$ OS is used.

7.3.1. Deploying cost

Every execution step has a cost associated in terms of GAS. Technically. GAS is the crypto-fuel of the Ethereum Virtual Machine (EVM). Fig. 9. shows the GAS consumed by the proposed smart contract when deploying it. The REMIX IDE provides five account

Table 4
Growth of blockchain-based PSCM network.

T_x	Per Minute (MB)	Per Hour (GB)	Per Day (GB)	Per Month (TB)	Per Year (TB)
50	11.95	0.700	16.80	0.50	5.99
100	23.89	1.4	33.60	1.02	11.97
250	59.74	3.50	84.00	2.54	29.94
500	119.47	7.00	168.01	5.09	59.88
1000	238.95	14.00	336.02	10.17	119.77

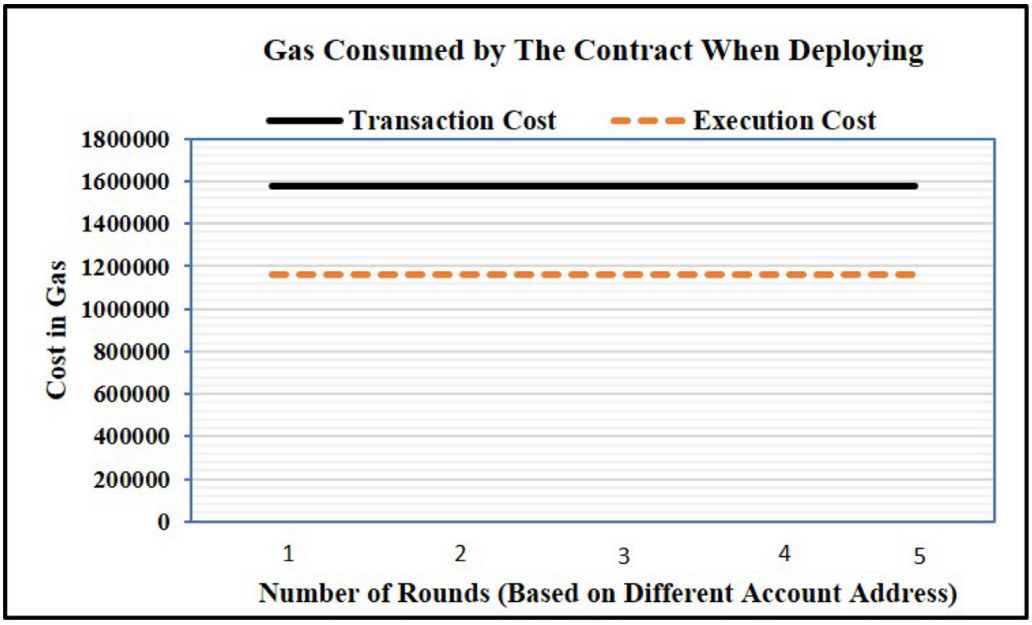

Fig. 9. Deploying cost of smart contract.

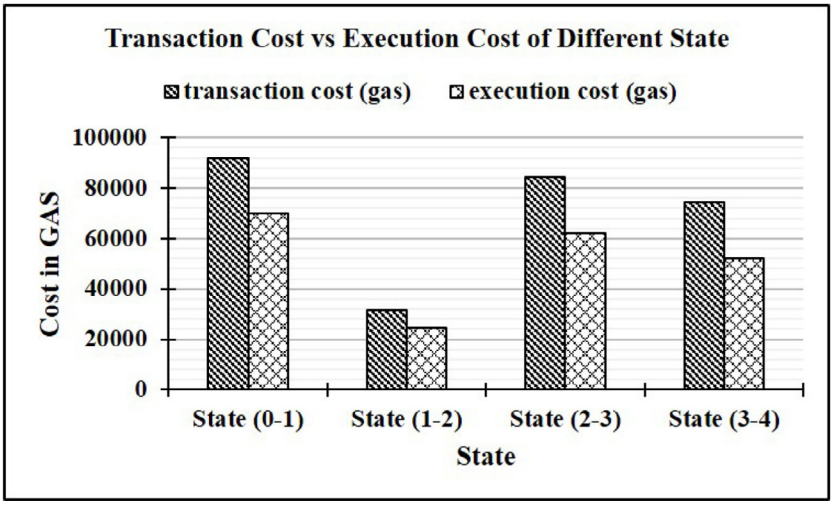

Fig. 10. Transaction cost and execution cost of smart contract.

addresses but, at a time only one address is used for deploying the smart contract. For a better understanding of deploying cost, all the five addresses are used in the different time-intervals. Surprisingly, the costs are the same, and the rate of growth of the curve is the same in the different intervals and linear too.

7.3.2. Transaction cost vs execution cost

Fig. 10. shows the analogy between the transaction cost and execution cost of all the states that are mentioned in the proposed state machine model-based smart contract. Transaction cost is the cost required for sending the code to the blockchain whereas, execution cost is the cost required to execute the computational operations. The result illustrated in Fig. 10. is the cost when the smart contract moves from one state to the next state and transaction costs are higher than the execution cost. For accurate analysis of the transaction and execution cost, the proposed smart contract runs using the different accounts (or addresses). But, interestingly, the costs are the same for all the account addresses.

7.3.3. Data manipulation with smart contract

The proposed smart contract is executed with a different number of inputs (number of products stored in the blockchain net-

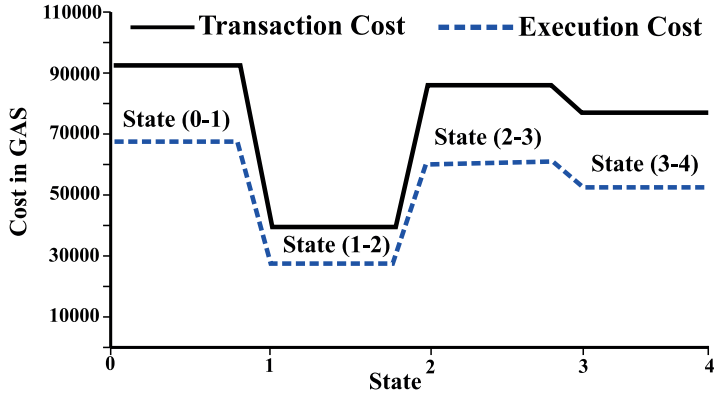

Fig. 11. Transaction cost and execution cost of smart contract with different numbers of inputs.

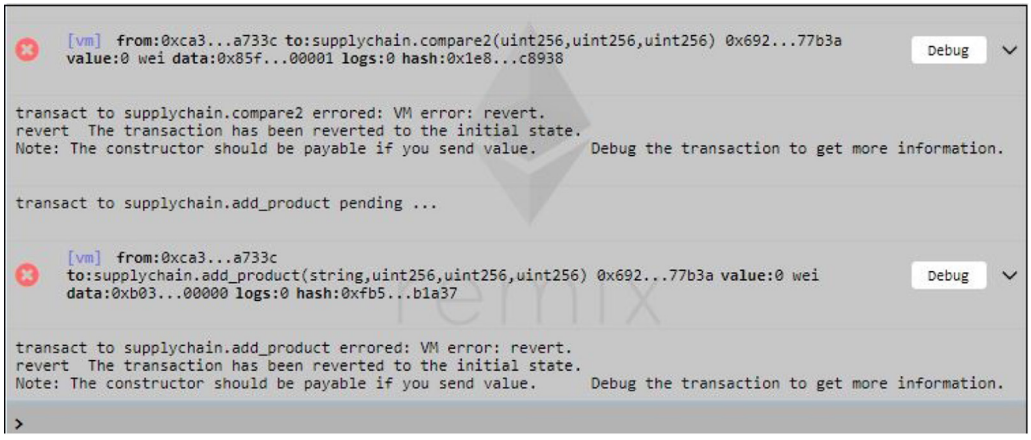

Fig. 12. Reverting to the initial state of the smart contract.

work) parameters. Fig. 11 shows the analogy between the transaction cost and execution cost with different inputs that are exchanged among states. The rate of growth of the curve is maximum as well as linear between state 0 to state 1 whereas, the rate of growth of the curve is minimum and linear between state 1 to state 2. For accurate analysis of the transaction and execution cost, the proposed smart contract runs with a different number of inputs that are accepted by each state. But, interestingly, the transaction cost and execution cost are the same and the rate of growth of the curve is linear for one state to the next state under the different numbers of inputs.

7.3.4. Reverting the state

Various transaction states such as manufacturer state, warehouse state, retailer state, etc. are used in the proposed smart contract. If any of the states want to check the details of any drug in the blockchain-based network, the drug details must be accessed through the smart contract. The smart contract checks the details of the product and it is based on the serial number of the product, manufacturing and expiry date of the product, at present which state is the owner of the product. If details of the product are correct then the product is handover to the next entity in the supply chain system otherwise the transaction is reverted to the initial state and an error message pop-up which is shown in Fig. 12.

8. Concluding remarks & future scope

In this paper, we present a blockchain-based approach for information sharing in the pharmaceutical supply chain management system. Our proposed system also provides a key management scheme with the update procedure in the smart-contract. Further, transaction and block validation protocols have been designed in our protocol. The security analysis confirms that the proposed protocol is secured against all possible security threats. We have made the performance of our protocol which shows that the protocol is not taking high computation and communication overheads. The evaluation of smart contracts in terms of transaction cost and execution cost is also discussed. Overall, our work provides various benefits to make the supply chain management better in comparison with the traditional system. Certain drugs are required to be transported at a particular temperature. The blockchain technology can provide the solution to this problem with the help of (IoT). In future research, we will be integrating the IoT and blockchain for better traceability of temperature-controlled drugs and improve the security and block verification 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.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.jisa.2020.102554.

CRediT authorship contribution statement

Sanjeev Kumar Dwivedi: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation,

Writing - original draft, Writing - review & editing, Supervision, Project administration. **Ruhul Amin:** Conceptualization, Methodology, Software, Validation, Investigation, Writing - original draft, Writing - review & editing, Supervision. **Satyanarayana Vollala:** Conceptualization, Software, Investigation, Data curation, Writing - review & editing.

References

[1] Westerkamp M, Victor F, Kupper A. Tracing manufacturing processes using blockchain-based token compositions. Digital Commun Netw 2019:1–10.
[2] Dabbene F, Gay P, Tortia C. Traceability issues in food supply chain management: a review. Biosyst Eng 2014;120:65–80.
[3] Kumari S, Renuka K. Design of a password authentication and key agreement scheme to access e-healthcare services. Wirel Pers Commun 2019:1–19.
[4] Al Omar A, Bhuiyan MZA, Basu A, Kiyomoto S, Rahman MS. Privacy-friendly platform for healthcare data in cloud based on blockchain environment. Future Gener Comput Syst 2019;95:511–21.
[5] Ali R, Pal AK, Kumari S, Karuppiah M, Conti M. A secure user authentication and key-agreement scheme using wireless sensor networks for agriculture monitoring. Future Gener Comput Syst 2018;84:200–15.
[6] Kumari S, Renuka K. A provably secure biometrics and ECC-based authentication and key agreement scheme for WSNs. Int J Commun Syst 2019;33:e4194.
[7] Kumari S, Li X, Wu F, Das AK, Arshad H, Khan MK. A user friendly mutual authentication and key agreement scheme for wireless sensor networks using chaotic maps. Future Gener Comput Syst 2016;63:56–75.
[8] Kumari S, Das AK, Wazid M, Li X, Wu F, Choo K-KR, et al. On the design of a secure user authentication and key agreement scheme for wireless sensor networks. Concurrency Comput 2017;29(23):e3930.
[9] Lin I-C, Hsu H-H, Cheng C-Y. A cloud-based authentication protocol for RFID supply chain systems. J Netw Syst Manage 2015;23(4):978–97.
[10] Sidorov M, Ong MT, Sridharan RV, Nakamura J, Ohmura R, Khor JH. Ultra-lightweight mutual authentication RFID protocol for blockchain enabled supply chains. IEEE Access 2019;7:7273–85.
[11] Shamshad S, Mahmood K, Kumari S. Comments on a multi-factor user authentication and key agreement protocol based on bilinear pairing for the internet of thingsg. Wirel Pers Commun 2020:1–4.
[12] Wang EK, Liang Z, Chen C-M, Kumari S, Khan MK. PoRX: a reputation incentive scheme for blockchain consensus of IIoT. Future Gener Comput Syst 2020;102:140–51.
[13] Azzi R, Kilany R, Sokhn M. The power of a blockchain-based supply chain. Comput Ind Eng 2019;135:582–92.
[14] Arzu Akyuz G, Erman Erkan T. Supply chain performance measurement: a literature review. Int J Prod Res 2010;48(17):5137–55.
[15] Steiner J., Baker J.. Blockchain: the solution for transparency in product supply chains. 2015. 2016.
[16] Abeyratne SA, Monfared RP. Blockchain ready manufacturing supply chain using distributed ledger. Int J Res Eng Technol 2016;5:1–10.
[17] Korpela K, Hallikas J, Dahlberg T. Digital supply chain transformation toward blockchain integration. In: Proceedings of the 50th Hawaii international conference on system sciences; 2017. p. 4182–91.
[18] Ruta M, Scioscia F, Ieva S, Capurso G, Di Sciascio E. Supply chain object discovery with semantic-enhanced blockchain. In: Proceedings of the 15th ACM conference on embedded network sensor systems. ACM; 2017. p. 60.
[19] Tönnissen S, Teuteberg F. Analysing the impact of blockchain-technology for operations and supply chain management: an explanatory model drawn from multiple case studies. Int J Inf Manage 2020;52(101953):1–10.
[20] Bocek T, Rodrigues BB, Strasser T, Stiller B. Blockchains everywhere-a use-case of blockchains in the pharma supply-chain. In: 2017 IFIP/IEEE Symposium on integrated network and service management (IM). IEEE; 2017. p. 772–7.
[21] Helo P, Hao Y. Blockchains in operations and supply chains: a model and reference implementation. Comput Ind Eng 2019;136:242–51.
[22] Schmidt CG, Wagner SM. Blockchain and supply chain relations: a transaction cost theory perspective. J Purch Supply Manage 2019;25(4):100552.
[23] Hughes L, Dwivedi YK, Misra SK, Rana NP, Raghavan V, Akella V. Blockchain research, practice and policy: applications, benefits, limitations, emerging research themes and research agenda. Int J Inf Manage 2019;49:114–29.
[24] Kshetri N. 1 Blockchain's roles in meeting key supply chain management objectives. Int J Inf Manage 2018;39:80–9.
[25] Zheng Z, Xie S, Dai H-N, Wang H. Blockchain challenges and opportunities: a survey. Int J Web and Grid Serv 2018;14(4):352–75.
[26] Saha A, Amin R, Kunal S, Vollala S, Dwivedi SK. Review on "blockchain technology based medical healthcare system with privacy issues". Secur Privacy 2019;5(2):01–14.
[27] Yli-Huumo J, Ko D, Choi S, Park S, Smolander K. Where is current research on blockchain technology a systematic review. PLoS ONE 2016;11(10):e0163477.
[28] Li X, Jiang P, Chen T, Luo X, Wen Q. A survey on the security of blockchain systems. Future Gener Comput Syst 2020;107:841–53.
[29] Alhadhrami Z, Alghfeli S, Alghfeli M, Abedlla JA, Shuaib K. Introducing blockchains for healthcare. In: 2017 International conference on electrical and computing technologies and applications (ICECTA). IEEE; 2017. p. 1–4.
[30] Xu L, Shah N, Chen L, Diallo N, Gao Z, Lu Y, et al. Enabling the sharing economy: Privacy respecting contract based on public blockchain. In: Proceedings of the ACM workshop on blockchain, cryptocurrencies and contracts. ACM; 2017. p. 15–21.
[31] Mohanta BK, Jena D, Panda SS, Sobhanayak S. Blockchain technology: a survey on applications and security privacy challenges. Internet Things 2019;8:100–7.
[32] Magazzeni D, McBurney P, Nash W. Validation and verification of smart contracts: aresearch agenda. Computer 2017;50(9):50–7.
[33] He X, Qin B, Zhu Y, Chen X, Liu Y. SPESC: a specification language for smart contracts. In: 2018 IEEE 42nd Annual computer software and applications conference (COMPSAC), vol. 1. IEEE; 2018. p. 132–7.
[34] Atzei N, Bartoletti M, Cimoli T. A survey of attacks on ethereum smart contracts (SoK). In: International conference on principles of security and trust. Springer; 2017. p. 164–86.
[35] Idelberger F, Governatori G, Riveret R, Sartor G. Evaluation of logic-based smart contracts for blockchain systems. In: International symposium on rules and rule markup languages for the semantic web. Springer; 2016. p. 167–83.
[36] Luu L, Chu D-H, Olickel H, Saxena P, Hobor A. Making smart contracts smarter. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. ACM; 2016. p. 254–69.
[37] Mingxiao D, Xiaofeng M, Zhe Z, Xiangwei W, Qijun C. A review on consensus algorithm of blockchain. In: 2017 IEEE International conference on systems, man, and cybernetics (SMC). IEEE; 2017. p. 2567–72.
[38] Sankar LS, Sindhu M, Sethumadhavan M. Survey of consensus protocols on blockchain applications. In: 2017 4th International conference on advanced computing and communication systems (ICACCS). IEEE; 2017. p. 1–5.
[39] Wang W., Hoang D.T., Xiong Z., Niyato D., Wang P., Hu P., et al. A survey on consensus mechanisms and mining management in blockchain networks. arXiv:180502707 2018;:1–33.
[40] Nguyen G-T, Kim K. A survey about consensus algorithms used in blockchain.. J Inf Process Syst 2018;14(1):101–28.
[41] Casino F, Dasaklis TK, Patsakis C. A systematic literature review of blockchain-based applications: current status, classification and open issues. Telemat Inf 2019;36:55–81.
[42] Hölbl M, Kompara M, Kamišalić A, Nemec Zlatolas L. A systematic review of the use of blockchain in healthcare. Symmetry 2018;10(10):470.
[43] S. Nakamoto. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. [Online]. Available: https://bitcoin.org/bitcoin.pdf.
[44] Dennis R, Owen G. Rep on the block: a next generation reputation system based on the blockchain. In: 2015 10th International conference for internet technology and secured transactions (ICITST). IEEE; 2015. p. 131–8.
[45] Fujimura S, Watanabe H, Nakadaira A, Yamada T, Akutsu A, Kishigami JJ. Bright: a concept for a decentralized rights management system based on blockchain. In: 2015 IEEE 5th International conference on consumer electronics-berlin (ICCE-Berlin). IEEE; 2015. p. 345–6.
[46] Chen S, Shi R, Ren Z, Yan J, Shi Y, Zhang J. A blockchain-based supply chain quality management framework. In: 2017 IEEE 14th International conference on e-business engineering (ICEBE). IEEE; 2017. p. 172–6.
[47] Kim HM, Laskowski M. Toward an ontology-driven blockchain design for supply-chain provenance. Intell Syst Account FinanceManage 2018;25(1):18–27.
[48] Figorilli S, Antonucci F, Costa C, Pallottino F, Raso L, Castiglione M, et al. A blockchain implementation prototype for the electronic open source traceability of wood along the whole supply chain. Sensors 2018;18(9):3133.
[49] Nakasumi M. for supply chain management based on block chain technology. In: 2017 IEEE 19th Conference on business informatics (CBI), vol. 1. IEEE; 2017. p. 140–9.
[50] Tian F. An agri-food supply chain traceability system for China based on RFID & blockchain technology. In: 2016 13th International conference on service systems and service management (ICSSSM). IEEE; 2016. p. 1–6.
[51] Lin C, He D, Huang X, Khan MK, Choo K-KR. DCAP: a secure and efficient decentralized conditional anonymous payment system based on blockchain. IEEE Trans Inf Forensics Secur 2020;15:2440–52.
[52] Chen J., Duan K., Zhang R., Zeng L., Wang W.. An ai based super nodes selection algorithm in blockchain networks. arXiv:180800216 2018;:01–13.
[53] Shrestha R, Bajracharya R, Shrestha AP, Nam SY. A new-type of blockchain for secure message exchange in VANET. Digit Commun Netw 2019:1–14.

Sanjeev Kumar Dwivedi is pursing Ph.D. in the Department of CSE from Dr. Shyama Prasad Mukherjee International Institute of Information Technology Naya Raipur, (DSPM-IIITNR), Chhattisgarh,India. He received M.Tech Degree in the Department of CSE from Pondicherry University, Puducherry, India, in 2013. His research interest includes VANET security, cryptography, and blockchain.

Ruhul Amin received Ph.D. in CSE from the Indian Institute of Technology (ISM) Dhanbad, Jharkhand, India, in 2017. Presently, he is working as an Assistant Professor in the Department of CSE, Dr. Shyama Prasad Mukherjee International Institute of Information Technology Naya Raipur (DSPM-IIITNR), Chhattisgarh, India. His research interest includes VANET security, authentication protocol, and blockchain.

Satyanarayana Vollala received Ph.D. in CSE from National Institute of Technology, Tiruchirappalli, (NITT) Tamilnadu, India, in 2017. Presently, he is working as an Assistant Professor in the Department of CSE, Dr. Shyama Prasad Mukherjee International Institute of Information Technology Naya Raipur (DSPM-IIITNR), Chhattisgarh, India. His research interest includes Hardware security and theoretical computer science.