Cluster Computing (2022) 25:2223–2240
https://doi.org/10.1007/s10586-021-03425-x

Design and implementation of transaction privacy by virtue of ownership and traceability in blockchain based supply chain

Mohit Mohit[1] · Sanmeet Kaur[1] · Maninder Singh[1]

Received: 19 January 2021 / Revised: 12 September 2021 / Accepted: 18 September 2021 / Published online: 2 November 2021

Abstract

Blockchain was at the top of the 2016 Gartner hype cycle and has been integrated into business profiles by numerous start-ups. Since the emergence of blockchain through Bitcoin, studies have been conducted to increase blockchain applications for nonfinancial uses. A supply chain is a sector where blockchain is anticipated to have crucial applications. In a traditional supply chain, maintaining traceability and ownership remains a serious issue. In the supply chain, blockchain can increase trust, improve traceability, and eliminate the middle man. It makes the supply chain more transparent though, raising the privacy issue. In this paper, a new approach for transaction privacy is proposed by considering ownership and traceability. The proposed system retains the advantages of blockchain and centralised database server. Its novelty lies in achieving privacy by generating symmetric keys, employing product codes and current timestamps, and it uses asymmetric key elliptic curve cryptography for transaction validation and user identification. The proposed system allows product owners to trace the product and enables its transfer. It protects the supply chain from counterfeit products. The Hyperledger Sawtooth blockchain was used for experiments. Security and privacy analysis show that the proposed system can afford privacy without impinging on traceability and ownership. The results estimate that privacy incorporation introduces an overhead of 4.4%. In the experiment, the performance of the proposed system bettered the results of the existing techniques such as POMS and b_verify.

Keywords Blockchain · Supply chain · Counterfeits · Privacy · Logistics

1 Introduction

Supply chains are an integral part of most businesses and are essential for the success of a company and customer satisfaction. Business success is inextricably linked to the supply chain performance of the business. According to a survey conducted by Deloitte, from 2014 [1], approximately 79% of organisations with superior supply chain capabilities (supply chain leaders) achieved substantial above-average growth in revenue. By contrast, only 8% of businesses with limited supply chain capabilities reported above-average growth. A supply chain strategy is critical to business success; however, the importance of the supply chain strategy is often underestimated, and hence it is ignored or receives less attention than other [2] operation areas. According to the 2012 report created by the Australian Securities and Investments Commission on corporate insolvencies, 44% of businesses in Australia failed due to poor strategic management [3]. During a medical emergency, supply chain performance can be a key differentiator between life and death. The burden on the global trade structures generated by COVID-19 outbreak emphasises an urgent requirement for global cooperation to sustain and improve the stability of international supply chains. Blockchain can mitigate the supply chain flaws exposed by COVID-19 and improve economic recovery [2]. IBM launched a blockchain network to overcome supply chain problems caused by COVID-19 [4].

✉ Mohit Mohit
mmohit_phd17@thapar.edu

Sanmeet Kaur
sanmeet.bhatia@thapar.edu

Maninder Singh
msingh@thapar.edu

[1] CSED, Thapar Institute of Engineering & Technology, P.O. Box 32, Bhadson Road, Patiala, Punjab 147004, India

Currently, the supply chain has become considerably complicated. Tracing back products and their parts to original suppliers is challenging; thus, eliminating defects becomes difficult. Supply chains are used in areas such as consumer goods, industrial equipment, and food products [5]. Friction in the supply chain is one of the most significant problems. Many to and from movements occur. An increase in uncertainty prevents the supply chain from working well. All stakeholders, that is, providers, suppliers, and clients, must deal with each other through central third-party entities. Consequently, stakeholders cannot deal directly with each other. Thus, a simple transaction turns into a lengthy procedure with many steps. In a traditional supply chain, only manufacturers can add data to the database; thus, proving ownership and tracing back product to the original supplier is challenging.

Blockchain could be the solution for many of these problems. Blockchain refers to a digital register that stores records in the form of many blocks [6]. For blockchain to function effectively, it must be spread over several nodes, which are normally computers. Because of its appealing characteristics, such as speed, transparency, and accessibility, blockchain attracts many users from industrial fields, specifically from supply chain industries. Blockchain can be extremely effective in ensuring the recording of all processes and assets that pass through a supply chain. Moreover, it can ensure that orders are tracked and payments are made in a transparent manner. The logistics industry can benefit considerably from blockchain in terms of transparency for customers and auditors and high levels of security. At times, although higher transparency can jeopardize the privacy of the system, a blockchain-based supply chain might cause privacy risks. Many researchers [7, 8] have been working to solve the privacy problems in blockchain based supply chain; however, to the best of our knowledge, the problem has not been addressed efficiently by any single solution. In a view of privacy, a blockchain-based supply chain system is designed in this paper. The proposed system utilises the blockchain where the tracing information is recorded at each step, and each participant can add the data for traceability and ownership after validation. In the proposed work, the privacy of transaction data is addressed with the inclusion of traceability and ownership.

Our contributions

The main contributions of this paper are:

(i) An approach is proposed to enhance privacy in the blockchain-based supply chain. The privacy of transaction data is implemented in such a way that traceability and ownership are not affected.

(ii) Blockchain-based algorithms are designed to create a product, to transfer product ownership, and to trace the product through the supply chain with the encrypted ledger.

(iii) Security and privacy analysis are performed to assess the privacy of data. The throughput and latency of this work are compared with the baseline (without privacy) blockchain to evaluate the performance.

The rest of this paper is structured as follows. Section 2 includes the literature published on blockchain-based supply chains and privacy types and their problems. Section 3 presents details about blockchain, its working, consensus, and blockchain-based corporate projects. Section 4 describes the properties of the Hyperledger Sawtooth blockchain. Section 5 focuses on operational requirements, supply chains, algorithm designs, data submission processes, security and privacy analysis. Section 6 presents the evaluation of the proposed system. Section 7 presents conclusions.

2 Related work

Traditional supply chain systems have been subjected to forgery and fraud because of data replicas. The blockchain is suitable for this purpose because it secures the data by storing it in secure repositories. Because of the expressive characteristics of blockchain such as transparency, ease of traceability, and auditability, it is used in different supply chains, especially pharmaceuticals, agriculture, and food industries. Modum.io (2017) [9], a start-up, applied Internet of Things (IoT) sensor devices with blockchain to secure the integrity of information and allow public access to temperature records. Modium.io implemented the Ethereum blockchain network for the pharmaceutical supply chain to maintain data immutability. However, transaction data privacy is not supported by the Modum.io.

Toyoda et al. [10] created an Ethereum-based blockchain by leveraging the idea of 'proof of possession' of Bitcoin to manage product ownership. The authors implemented a complete protocol, which allows each entity of the supply chain to transfer and confirm product ownership. However, authors have not designed the protocol of privacy of transaction data.

Malik et al. [11] presented a case of adoption of the blockchain technology and IoTs in the agriculture food chain supplies to ensure consumer protection; that is, optimal practices in food handling and processing. The authors presented the framework *ProductChain*, which confirms the accessibility of data to users, restricts the entry of competitive business partners, and provides confidential trade flows between consumers and stakeholders. Authors claims the privacy using asymmetric key encryption but privacy enabled algorithms and validation steps are missing.

Caro et al. [12] proposed a model to harmonise IoT and blockchain into the agricultural food supply chain through *AgriBlockIoT*. The *AgriBlockIoT* system architecture is expansive from providers to producers, distributors, retailers, and consumers. This architecture operation is backed up by application programming interface (API) that seamlessly blends IoT and blockchain into agricultural food chain supply. Authors have not proposed any algorithm for privacy of data.

Su et al. [13] presented *SmartSupply* blockchain, based on the Ethereum platform, by focusing the organising validator's data structure for information retrieval and transaction validation. The *SmartSupply* system delivers uniform time latency per query irrespective of the blockchain size. Authors well defined the supply chain, and miners' validation steps but privacy of transaction is not mentioned.

Leng et al. [14] introduced the double-chain blockchain structure for an agricultural business to increase privacy and transparency. The first blockchain is applied to preserve the user data; the second blockchain is applied for transaction data. The proposed blockchain only provides privacy to user data. The proposed blockchain only provides privacy to user data; not transaction data.

Chod et al. [15] proposed Bitcoin-based protocol; b_verify aims to provide digitisation of invoices to improve financial support for the agricultural supply chain. b_verify leverages the use of public blockchain and ensures immutability at low cost, however authors have not addressed the issue of privacy. Malik et al. [16] introduced TrustChain that leverages the use of consortium blockchain to monitor interactions between supply chain members and to dynamically allocate reputation scores based on these interactions. The TrustChain can defend the sybil, repudiation of transaction, and impersonation attacks, however privacy leakage attack is possible.

Lin et al. [17] presented an Ethereum-blockchain-based food traceability system with the IoT technology in the food chain supply to mitigate the risk of data explosion. To enhance performance and decrease the traceability cost, the proposed architecture uses an on- and off-chain data methodology. Authors claims that the proposed architecture provides the privacy, however privacy and validation algorithms have not been described.

Du et al. [7] designed a supply chain solution for the finance sector. They used a consortium blockchain to protect the privacy and implemented an acceptable homomorphic encryption. The complete homomorphic encryption is not feasible for blockchain. Fully homomorphic encryption has adequate privacy protection but is low in performance.

Shahid et al. [18] proposed a blockchain-based approach for addressing the issues of product traceability, trading party credibility, and delivery mechanism in the agri-food supply chain. All transactions are recorded on the proposed system's blockchain, which eventually uploads the data to the interplanetary file storage system (IPFS). The storage system generates a hash of the data that is then stored on the blockchain, ensuring an efficient, safe, and trustworthy solution. However, the issue of privacy remains unaddressed.

Liu and Li [19] developed a cross-border blockchain-based supply chain to protect against clone attack and counterfeit tag attack. Multichain architecture is used to store account data, transaction data and IoT data . The authors use ECC and RSA encryption to protect product tags from being counterfeit. However, transaction validation algorithms have not been mentioned.

Iqbal and Butt [20] proposed an IoTs-based agriculture system that uses IoT sensors to detect animal attacks. The blockchain has been used to store information about animal attacks and share the information with another node. The proposed system does not include data privacy as a feature.

Latif et al. [21] developed the blockchain-based supply chain to store all commodity history through smart contracts. The system can easily trace the source of goods and verify the parties' identity to ensure the transaction's legitimacy. However, the system does not support the privacy of data.

2.1 Corporate projects of blockchain based supply chain

Blockchain can revolutionise the supply chain by ensuring that the value of goods reflects the manufacturing cost of the goods. For instance, Walmart is testing blockchain for use in supply chain management [22]. Everledger provides a platform for identifying fraud [23]. A stable diamond ledger is the primary use case presented by Everledger. In the ledger, diamond verification is recorded on the blockchain and can be verified by law enforcers, owners, and insurance companies. According to a study, 67% of fraudulent insurance claims remain undetected [24]. Therefore, to prevent fraud, Everledger offers digital certificates, which is their primary goal. Everledger has a business-to-business service model, and the blockchain used by Everledger is private. To avoid counterfeit and forgeries, Blockverify a US-based company, is trying to bring blockchain into the supply chain [25]. The main businesses that it deals in luxury items, pharmaceuticals, electronics, and diamonds. In the pharmaceutical sector, a pilot project was conducted by Blockverify. According to an estimate by the International Policy Network, about 700,000 deaths per year are caused by fake malaria and tuberculosis drugs [26]. Blockverify uses a combination of Bitcoin blockchain and a private side chain. The private key of every product is stored in the public blockchain, which can be checked by anyone. A change in ownership can be easily traced with the help of a track and trace number, which is recorded in the private blockchain. Verisart, another start-up, is using blockchain to certify, document, and verify artwork [27]. To record data regarding the artwork in the blockchain, every artist can scan their artwork and provide metadata for the image identification algorithm. Moreover, people can track ownership and artist information for the artwork, such as information on who has the artwork and where its current location is. Sophiatx is creating its own blockchain for pharmaceutical industries. The aim of the Sophiatx blockchain is to eradicate the challenges of traceability in the supply chain when multiple parties are involved [28]. Provenance is using blockchain to examine the authenticity of a product and its origin, thereby creating transparency in the supply chain [29]. The information is stored and recorded in the blockchain without the need for intermediate auditors. With this setup, anyone can have a detailed view of the supply chain and obtain details regarding which part is created and assembled by whom. Chronicled is another blockchain company that focuses on luxury items and fraud prevention [30]. To link a product to the blockchain, it uses a protected smart tag. Information regarding sellers and buyers is stored in the blockchain, which acts as an open registry. Therefore, with the help of the tag, anyone can verify the old records of sellers and buyers.

The systematic literature review of the blockchain-based supply chain indicated that numerous studies investigated transparency and security. However, no definitive study has been conducted to achieve privacy. We described how the proposed system attains security and transparency with the inclusion of privacy.

3 Background

In the software world, every few years (now a few months), new disruptive technologies emerge that are promising and appear to solve all software problems. In 2016, one such technology known as blockchain rose to prominence. Blockchain was designed in 2008 for Bitcoin. In addition to Bitcoin, blockchain can be applied to diverse applications. A blockchain is a distributed and immutable ledger that allows the recording of transactions and the tracing of resources in a business network [31]. The transactions are stored in blocks. Once the transaction is recorded within the blockchain, it becomes difficult to change it. Every new transaction is validated across the distributed network before it is stored in a block. Each block is identified by its cryptographic signature. A pictorial representation of blockchain is displayed in Fig. 1.

Figure 2 illustrates the working of blockchain.

(i) When a node creates and publishes a transaction to the network.

(ii) The transaction is added to the list of unconfirmed transactions. While there is no single authoritative list, each member of the network maintains their own and distributes it to others with whom they are connected.

(iii) Each node in the blockchain network is aware of the validation rules that must be followed to validate the transaction. Invalid transactions are rejected, whereas valid transactions are propagated to linked nodes, which validate and forward the transaction to their peers until it reaches every node in the network.

(iv) The transactions that occur over a specified time are stored in a block. After validation, the block is broadcasted to all the nodes in the network for synchronization. If the entire network comes to a consensus and all nodes accept the new block, it is chained into the blockchain.

(v) Once enough subsequent blocks confirm a recorded transaction, it becomes an immutable part of the ledger.

3.1 Consensus

A consensus algorithm is a mechanism, through which all peers of the blockchain network reach a shared agreement on the current state of the distributed ledger. Agreeing on transaction validity before the integration of transaction into blockchain is mandatory for blockchain network members. All new changes must be assessed and confirmed before incorporation. Obtaining consensus in a blockchain network guarantees that all nodes in the network agree upon a constant global state of blockchain [32]. Consensus

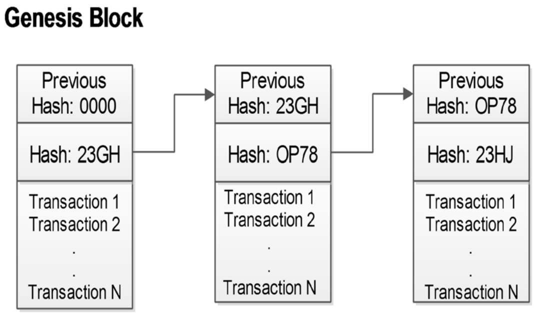

Fig. 1 Blockchain with hash values

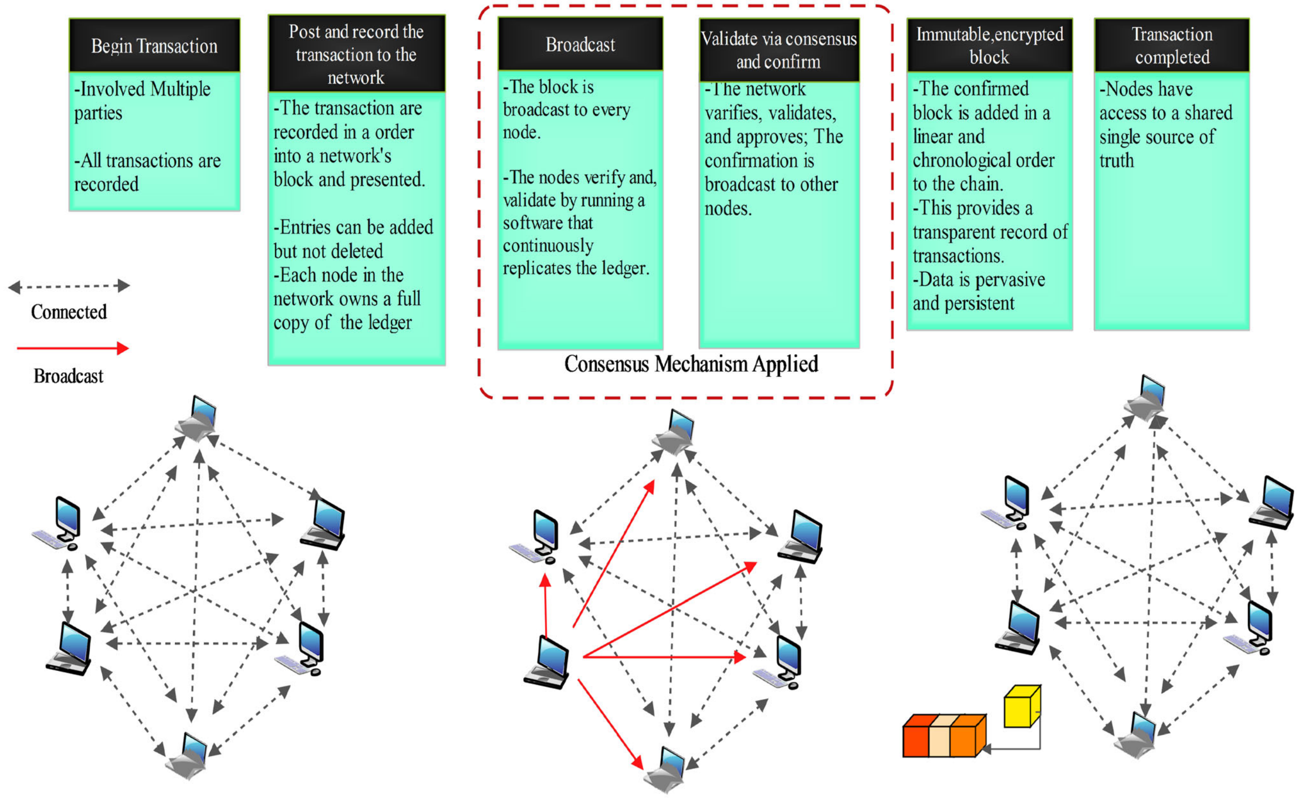
Begin Transaction
-Involved Multiple parties
-All transactions are recorded
Post and record the transaction to the network
-The transaction are recorded in a order into a network's block and presented.
-Entries can be added but not deleted
-Each node in the network owns a full copy of the ledger
Broadcast
-The block is broadcast to every node.
-The nodes verify and, validate by running a software that continuously replicates the ledger.
Validate via consensus and confirm
-The network varifies, validates, and approves; The confirmation is broadcast to other nodes.
Immutable,encrypted block
-The confirmed block is added in a linear and chronological order to the chain.
-This provides a transparent record of transactions.
-Data is pervasive and persistent
Transaction completed
-Nodes have access to a shared single source of truth
Consensus Mechanism Applied
Connected
Broadcast

Fig. 2 Working of blockchain

characteristics involve distributed governance, integrity, authentication, non-repudiation, a quorum structure (to exchange messages between nodes in a predefined manner), fault tolerance of Byzantine, and efficiency.

Proof of Work, PoW [33] was the first cryptocurrency consensus protocol that enabled Bitcoin participants to achieve consensus. The construction of each new block requires the miner to resolve a cryptographic puzzle, and the miner who first resolves the puzzle transmits the outcome to the network and receives the reward. The Proof of Stake, PoS [34] consensus algorithm selects the miner to mine the requested block on the ground of miner's economic stake. In the PoS algorithm, all miners deposit their stake to gain opportunities to mine the next block. The greater the stake, the greater are the opportunities. Although it is not asserted that a miner with the highest stake would be chosen, the algorithm selects the miner randomly similar to a lottery. Proof of Elapsed Time (PoET) is an energy-efficient consensus algorithm. In PoET, a random delay time T is allocated to each validator. The validator whose time expires first is selected as a miner for the next block [34]. In distributed computing, the practical Byzantine fault tolerance (PBFT) algorithm [35] expands and resolves the classical issue of Byzantine generals. PBFT is developed for the permissioned blockchain to consume the minimum computing (or hash) energy. Each node in the network retains its inner state, and when a message is obtained, it performs computation and makes a decision on the received message. Each node sends its decision to the leader, which affirms the trust of the received message. The leader believes that out of N total nodes, F nodes at most can be defective, where $F < N/3$.

3.2 Privacy problem in blockchain

Blockchain provides transparency in the transaction, and blockchain ledger data is viewable to all participants. This feature can escalate the risk of privacy leakage; an attacker can conveniently expose the product information and sales data. By default privacy of content is not the feature of blockchain. To protect the privacy in the supply chain, blockchain needs to satisfy the following requirements.

(i) The linking between the transaction must be traceable.

(ii) The transaction data must be encrypted. Only the authorised user or owner of transaction can decrypt it.

If a transaction with encrypted data is requested for validation, then the validator cannot validate the transaction. Hence, the transaction cannot be appended to the blockchain. Many researchers are paying attention to privacy protection, and some approaches have been attempted in the blockchain [36]. The main approaches are the blind signature, ring signature, homomorphic encryption and zero-knowledge proof. These methods provide privacy at the cost of traceability [37]. These methods are good for healthcare and voting applications, where traceability is not a desirable feature. In a supply chain, however, traceability is desirable.

4 Choice of blockchain platform

Wang et al. [38] conducted an analysis of the four most widely used blockchain platforms, namely Ethereum, Fabric, Sawtooth, and Fisco-Bcos. They concluded that Hyperledger Sawtooth and Fisco-Bcos outperform Hyperledger fabric and Ethereum in terms of performance (latency and throughput). Hyperledger Sawtooth's architecture is simple and modular, with plenty of room for customization. Caro et al. [12] assessed Ethereum and Hyperledger Sawtooth blockchain platforms on the basis of latency, network traffic and CPU load and they observed that Hyperledger Sawtooth performs better than Ethereum. In the proposed work, Hyperledger Sawtooth blockchain has been used. The Hyperledger Sawtooth provides the option to use permissionless and permissioned blockchain and it offers consensus algorithm PoET which could be better suited for tiny machines.

Hyperledger Sawtooth is an open source and enterprise blockchain platform that can be used to implement the distributed ledger networks and develop applications. The key feature of Hyperledger Sawtooth is that it is rather modular, which facilitates enterprises to select their own transaction rules, consensus algorithms, and permission systems catering to their respective business requirements [39]. Figure 3 illustrates the architecture of Hyperledger Sawtooth.

The following are the main components of Hyperledger Sawtooth.

(i) Validator is the central element of Sawtooth blockchain node. It communicates with transaction processor, consensus engine and REST API. The validator is responsible for peer-to-peer communication.

(ii) Clients program create and submit the transaction to validator via REST API.

(iii) REST API allowing clients to use common HTTP/JSON standards to interact with a validator. Sawtooth has pragmatic REST API that provides a language independent interface for submitting and reading transactions. The REST API treats the validator as a black box to send transactions and get results.

(iv) Business logics are defined in the transaction processors, The transaction processor has power to accept and reject the transaction. Based on the decision taken by transaction processor, the validator accept or abandon the transaction.

(v) Consensus engines allow Sawtooth to get more consensus option. Consensus engine is not the part of validator, and it works as a separate process such as REST API and transaction processors.

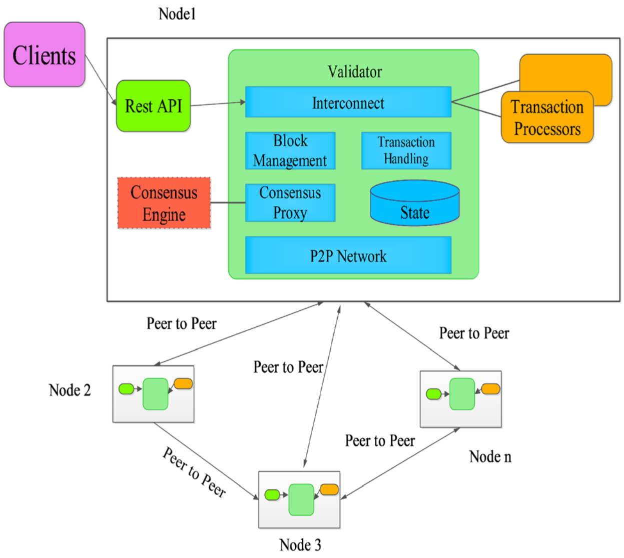

Fig. 3 Hyperledger Sawtooth architecture [40]

5 Proposed system

The centralised system is good for privacy, but it is not good for proving product ownership. Although the blockchain is useful for proving ownership, it lacks privacy. The proposed design incorporates elements of blockchain and centralised systems. The proposed design leverages the blockchain's and centralised database system's properties. The proposed system can be used for products that are prone to counterfeiting, such as protein supplements. Protein products are expensive, and the likelihood of counterfeiting them is also high. Generally, protein manufacturers provide an online facility for verifying the authenticity of the bar code on the product; however, because a single bar code can be used multiple times, so establishing ownership is critical. Our proposed system prevents the product from being counterfeit, traces the provenance, and maintains the product's privacy. In the proposed system, the manufacturer is responsible for creating the product and its symmetric key. Each product has a distinct symmetric key. The product's transaction data is encrypted using its symmetric key; the transaction data is stored on the blockchain, while the product's key is stored on the centralised database server (CDS). The manufacturer uses the client application to generate the transaction for product creation on the blockchain. The client application uses Algorithm 3 to create the product and attach the manufacturer's public key with the transaction. The proposed system believes that any product that a manufacturer creates is original. The validator only verifies the transaction of product creation is coming from the manufacturer or not. Once the validator accepts the transaction, the product is added to the blockchain. To transfer the product, the user must be the owner of the product. To transfer the product, the client application uses Algorithm 6. The validator uses Algorithm 7 to verify the transaction. The process flow of product transfer is illustrated in the Fig. 8. The participant who possesses the product can verify it using Algorithm 8.

The privacy and validation algorithms are designed in such a way that the symmetric key is accessible only to a legitimate validator or product owner. The CDS can be a single point of failure in the design; however, the risk of the CDS failing can be mitigated by creating multiple replicas of the CDS on premise (geographically dispersed) or on cloud.

The proposed system in detail is addressed in this segment. In particular, first, the essential system specifications are described, followed by the pseudo-codes of the smart contracts implemented in blockchain. Subsequently, algorithmic processes between all the entities required for the implementation of the proposed system are described.

5.1 Operational requirements

Before presenting the proposed approach, we describe the primary requirements and explain their importance for appropriate functioning.

(i) A manufacturer can create the product on blockchain and CDS.
(ii) The manufacturer and distributor can transfer the product.
(iii) The owner of the product can only initiate the transfer request.
(iv) The manufacturer, distributor, and customer can trace the product.
(v) Every participant of the supply chain will use the same blockchain.

A product can be identified using a barcode or RF-ID. The first requirement prevents the creation of a counterfeit product on blockchain. The second and third requirement states that only the legitimate user could transfer the product. The fourth requirement indicates that any actor who has the product can trace the product. In blockchain, each user is identified by their public key.

5.2 Supply chain

In the proposed system, a CDS and blockchain are used. Blockchain stores the encrypted data of the transaction, which can be decrypted using the symmetric key stored in the CDS. Figure 4 illustrates the operational steps of the proposed system. The supply chain mainly comprises five actors, namely manufacture, first distributor, second distributor, retailer, and consumer. Each product is recognised using a unique code (Bar, QR, or RF-ID), denoted as Pi.

Only manufacturer M1 can create the product. The manufacturer generates unique symmetric key SYM_KEYi by using product Pi (Bar, QR Code or RF-ID) and timestamp Ti, as described in Algorithm 3, and stores SYM_KEYi with SHA256 and SHA244 hashes of product Pi in the CDS as depicted in the step 1 and 2 of Fig. 4. For each product Pi, a unique SYM_KEYi is available. Manufacturer transfers the product Pi to distributor D1 as shown in step 3 of Fig. 4. The actor who receives product Pi can obtain SYM_KEYi by querying the CDS through REST API as illustrated in step 4 of Fig. 4. With the help of SYM_KEYi, the transaction records of Pi are decrypted and verified as shown in the step 5 of Fig. 4. To implement blockchain, Hyperledger Sawtooth is used. To implement CDS, a combination of MySQL, memcache, and REST API is used. Memcache stores all the data in RAM, which saves the IO operations; consequently, the response of REST API enhances. Figure 5 illustrates the architecture of the proposed system. Mainly, three entities are described in the architecture, namely CDS, blockchain node of the manufacturer, and other participants. The CDS stores the SYM_KEY of each product and provides it to the validator and product owner upon querying. Each blockchain node comprises two essential components, namely data submitter and validator. The data submitter allows a user to submit a transaction, and the validator is responsible for transaction generation and validation. The architecture design only allows the manufacturer node to create the transaction of product creation on blockchain.

5.3 Smart-contracts (rules) implementation

For privacy, symmetric key encryption is used. All keys are stored in the CDS, which can be retrieved by querying REST API. For product Pi, two types of transactions exist.

(i) Incoming transaction (Ti),
(ii) Stored transaction (Ts).

If product Pi is created the first time, then Ts does not exist. A transaction contains four data items, including SHA244

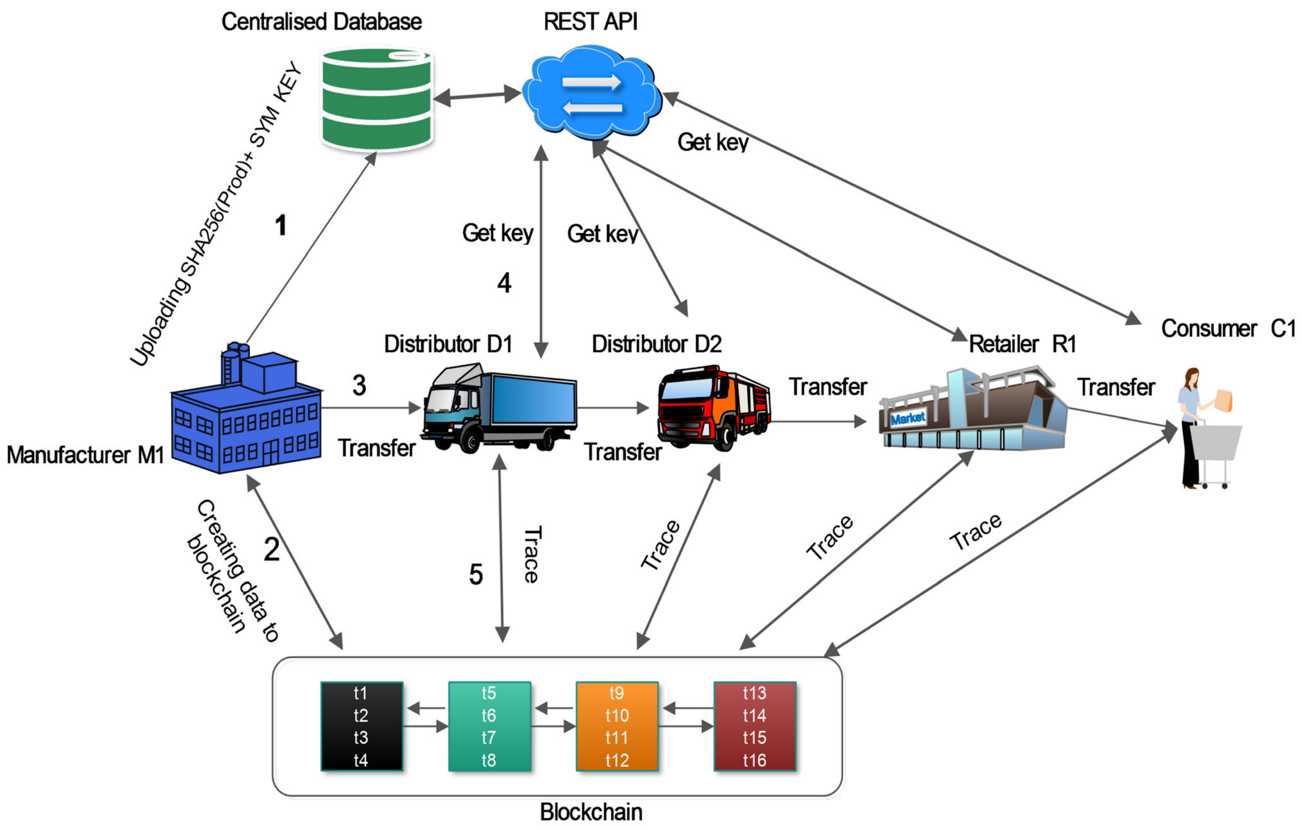

Fig. 4 Supply chain

hash of product, action, encrypted data, and timestamp as illustrated in Fig. 6.

The encrypted data contains Owner1 and Owner2, which can be revealed after decryption by using SYM_KEYi. For Ti, Owner1 and Owner2 are new and current owners, respectively. Similarly, for Ts, Owner1 and Owner2 represent current and old owners, respectively. Based on rules, the validator rejects or accepts the transaction. The algorithms for the proposed system are presented in detail.

Algorithm 1 Encrypt_data() //Encrypts the data. *Encrypt()* uses the AES.

```
Inputs: SYM_KEYi of product Pi, data
Output : Enc_Cipher_data
En_data = Encrypt(data, SYM_KEYi)
Base64_En_data = Base64.encode(En_data)
Return Base64_En_data
End
```

Algorithm 1 uses the symmetric key SYM_KEYi of Pi to encrypt the data. Algorithm 3 defines the creation of SYM_KEYi. *Encrypt()* uses an AES encryption system and returns the encrypted data denoted as *En_data*. *En_data* is further encoded using Base64 encoding scheme to preserve the data in the ASCII format.

Algorithm 2 Decrypt_data() //Decrypts the data. Decrypt() uses the AES

```
Inputs: SYM_KEYi of product, and Base64_En_data
Output : data
En_data = b64decode(Base64_En_data)
data = Decrypt(En_data,SYM_KEYi)
Return data
End
```

Algorithm 2 defines the decryption process. It takes SYM_KEYi and data denoted as *Base64_En_data*, which is returned through Algorithm 1. Algorithms 1 and 2 are base algorithms for encryption and decryption, which are used by forthcoming algorithms.

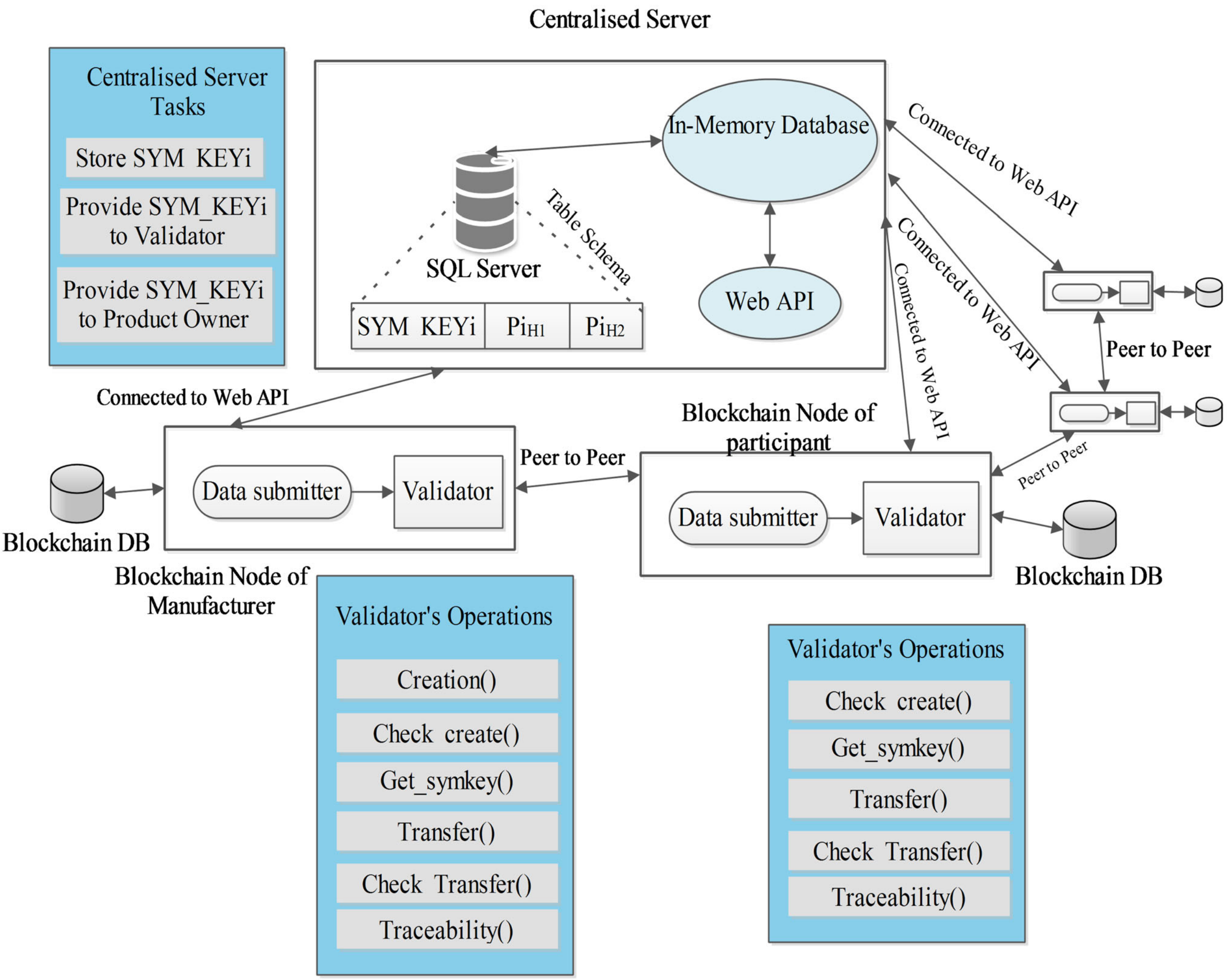

Fig. 5 Proposed system architecture

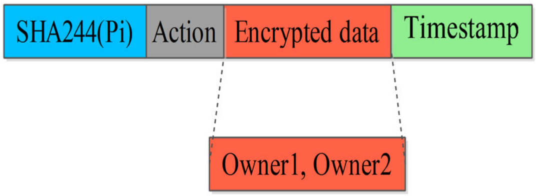

Fig. 6 Transaction data

Algorithm 3 Creation() //Creates a product on blockchain

```
Inputs: Pi, $M1_{PK}$
Output : $Pi_{H2}$, Action, Encypted_data, Ti
Generate current Unix time stamp Ti
$Ti_F$ = Ti in 6 decimal digit of precision
$Ti_d$ = Ti in integer form.
PiTi = Pi || $Ti_F$
SYM_KEYi = SHA256(PiTi)
$Pi_{H1}$ = SHA256(Pi)
$Pi_{H2}$ = SHA244(Pi)
Store the SYM_KEYi with $Pi_{H1}$, $Pi_{H2}$ to the CD
Data = $M1_{PK}$ + $M1_{PK}$
Encypted_data = Encrypt_data(SYM_KEYi, Data)
Action = "Create"
Return $Pi_{H2}$, Action, Encypted_data, $Ti_d$
End
```

Algorithm 3 describes the transaction (T_{CR}) of product creation. It takes product code Pi and manufacturer's public key $M1_{PK}$. Pi and the current Unix timestamp are unique numbers. Therefore, their concatenation also produces a unique string PiTi. The SHA256 hash of PiTi must be unique; consequently, it can act as a symmetric key denoted as SYM_KEYi. The symmetric key generation algorithm must be kept a secret that is only known to the manufacturer. The SYM_KEYi, hash SHA256 and SHA244 of Pi are stored on the CDS. The data that must be encrypted, comprises the public key of current and new owners, and in this scenario both are the same. *Encrypt_data()*, defined in Algorithm 1, encrypts the data; finally, Pi_{H2}, action, Encypted_data, and Ti form the transactional data.

Algorithm 4 Check_create() // Called by validator to validate the transaction

```
Inputs: signer_public_key
Ouput: True or False
if Action=="Create" then
    if Ts of Pi not exists and signer_public_key == $M1_{PK}$ then
        Return True
    else
        Return False
End
```

Algorithm 4 implemented on a validator machine. If the 'action' field of the transaction shows 'create' then the validator calls the check_create() of Algorithm 4. Check_create() only works if product Pi is not already created and the manufacturer has initiated the transaction. The manufacturer's public key $M1_{PK}$ is hard-coded in the software. Check_create() matches the transaction signer's key with hard-coded $M1_{PK}$ if match found then transaction accepted; rejected otherwise. Figure 7 illustrates the steps of the process flow of product creation and validation.

Algorithm 5 Get_symkey() // Called by Validator to get the sym_key

```
Inputs: $V_{pub\_key}$ , $Pi_{H2}$
Output : SYM_KEYi
Call the REST API URL with $V_{pub\_key}$ and $Pi_{H2}$
if $V_{pub\_key}$ in the V_list then
    get the SYM_KEYi of $Pi_{H2}$ from CDS
    $SYM\_KEYi_{en}$ =ECC_encrypt($V_{pub\_key}$, SYM_KEYi)
    Return $SYM\_KEYi_{en}$
else
    Return None
End
```

The validator calls Algorithm 5 to obtain the SYM_KEYi of Pi because for validation of the transaction, the validator requires the SYM_KEYi; consequently, the validator calls REST API with its public key (V_{pub_key}) and product Pi. The CDS maintains a list of public keys (V_list) of all validators; if the public key exists in the V_list, then the CDS returns SYM_KEYi_{en}, which is the public key ECC form of SYM_KEYi. The validator with V_{pub_key} can only decrypt SYM_KEYi_{en} to SYM_KEYi by using its private key.

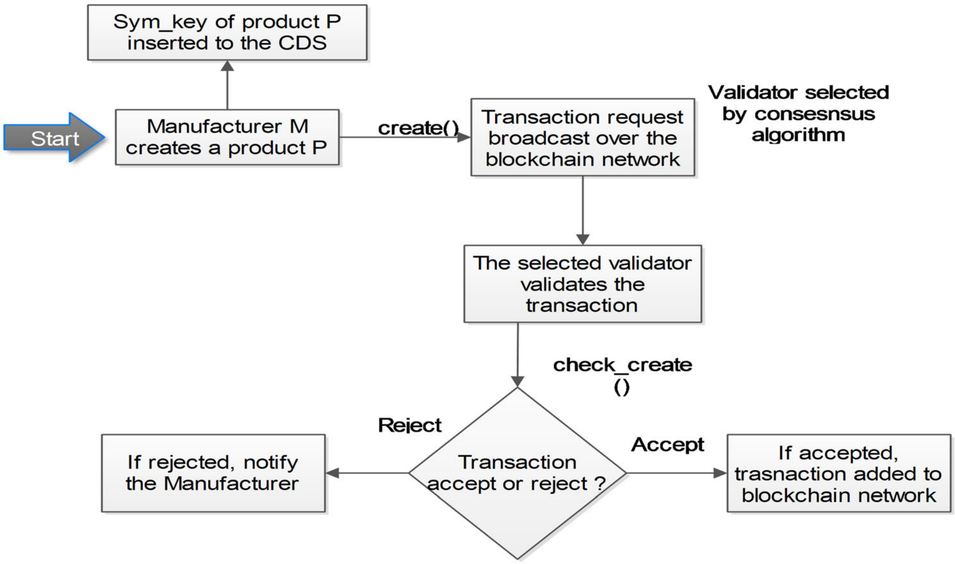
Sym_key of product P inserted to the CDS
Start
Manufacturer M creates a product P
create()
Transaction request broadcast over the blockchain network
Validator selected by consesnsus algorithm
The selected validator validates the transaction
check_create ()
Reject
Accept
Transaction accept or reject ?
If rejected, notify the Manufacturer
If accepted, trasnaction added to blockchain network

Fig. 7 Process flow of product creation

Algorithm 6 Transfer() //Called when the owner of product transfer the ownership to new owner

```
Inputs: $CO_{PK}$, Pi and $NO_{PK}$
Output : $Pi_{H2}$, Action, Encypted_data, Unix timestamp
$Pi_{H2}$ = SHA244(Pi)
$Pi_{H1}$ = SHA256(Pi)
Retrieve SYM_KEYi by calling REST API of CDS with $Pi_{H1}$
Data = $CO_{PK} + NO_{PK}$
Generate current Unix time stamp
Encypted_data = Encrypt_data(SYM_KEYi, Data)
Action = "transfer"
Return $Pi_{H2}$, Action, Encypted_data, Unix timestamp
End
```

Algorithm 6 creates a transaction (T_{TR}) that allows transfer of product ownership. Algorithm 6 uses the public key (CO_{PK}) and product code (Pi) of the current owner and the public key (NO_{PK}) of the new owner as an input. It retrieves the SYM_KEYi of Pi, created at the time of product creation, from the CDS. With the help of Algorithm 1, the data that comprises CO_{PK} and NO_{PK} are encrypted, which is denoted as *Encypted_data*. Finally, SHA244 hash of Pi Pi_{H2}, 'transfer' as action, *Encypted_data*, and current unix timestamp are requested for the transaction.

Algorithm 7 Check_Transfer() // Called by Validator to validate the transaction

```
Inputs: $Pi_{H2}$, Action, Encypted_data, Unix time stamp,$V_{pri\_key}$, $V_{pub\_key}$
Output : True or False
$SYM\_KEYi_{en}$ =Get_symkey($Pi_{H2}$,$V_{pub\_key}$)
SYM_KEYi = ECC_Decrypt($V_{pri\_key}$ , $SYM\_KEYi_{en}$)
Data = Decrypt_data(SYM_KEYi, Encypted_data)
Retrieve $CO_{PK}$, $NO_{PK}$ from Data
if Action=='transfer' and Pi is already created then
    get latest Ts of Pi from the ledger
    if Owner1 of Ts = $CO_{PK}$ then
        Return True
    else
        Return False
End
```

The selected validator calls Algorithm 7 to validate the transaction. Algorithm 7 takes transaction data as an input. The validator does not know Pi; thus, they call the Get_symkey (Pi_{H2},V_{pub_key}) and retrieve the SYM_KEYi_{en} of Pi. The SYM_KEYi_{en} is further decrypted using ECC_Decrypt(V_{pri_key}, SYM_KEYi_{en}). ECC_Decrypt() demonstrates the ECC decryption process. To decrypt *Encypted_data*, Algorithm 2 is called; after decryption, the public keys of the current owner CO_{PK} and new owner NO_{PK} are retrieved. If action is 'transfer,' then Pi must be already created on blockchain. If CO_{PK} matches with Owner1 of latest Ts, then the validator accepts the transaction, otherwise rejects.

Algorithm 8 Traceability() //Called when an owner of a product or customer want to perform traceability.

```
Inputs: Pi
Output : Mi and Owner
$Pi_{H2}$ = SHA244(Pi)
$Pi_{H1}$ = SHA256(Pi)
Select all the transactions which contain $Pi_{H2}$
Retrieve SYM_KEYi by calling REST API of CDS with $Pi_{H1}$
for all selected transactions do
    Extract Encypted_data field
    Data = Decrypt_data(SYM_KEYi, Encypted_data)
    Retrieve owner1, owner2 from Data
    if owner1 of previous transaction == owner2 of new transaction then
        Return owner2 of first and last transactions.
    else
        Return False
End
```

Anyone who owns the product can call Algorithm 8 (Traceability()). Algorithm 8 takes product Pi and returns the origin and current owner. It selects all the transactions, which contain Pi_{H2}, and obtains SYM_KEYi from the CDS. With the help of SYM_KEYi, Encypted_data is decrypted. Traceability is performed on a local copy of the blockchain data by matching Owner1 of the previous transaction with Owner2 of the new transaction. The time complexity of Algorithm 8 (Traceability()) is O(N), where N is the total number of transactions. Figure 8 illustrates the steps involved in product transfer.

5.4 Data submission process

The following four operational phases are performed in the data submission process.

5.4.1 Data submission

The manufacturer can create the product on blockchain and CDS by invoking Algorithm 3. The SYM_KEYi generation procedure must be kept secret and run by the manufacturer. The manufacturer and distributors can transfer the product by invoking transfer(). The party who owns the product can obtain SYM_KEYi by calling REST API.

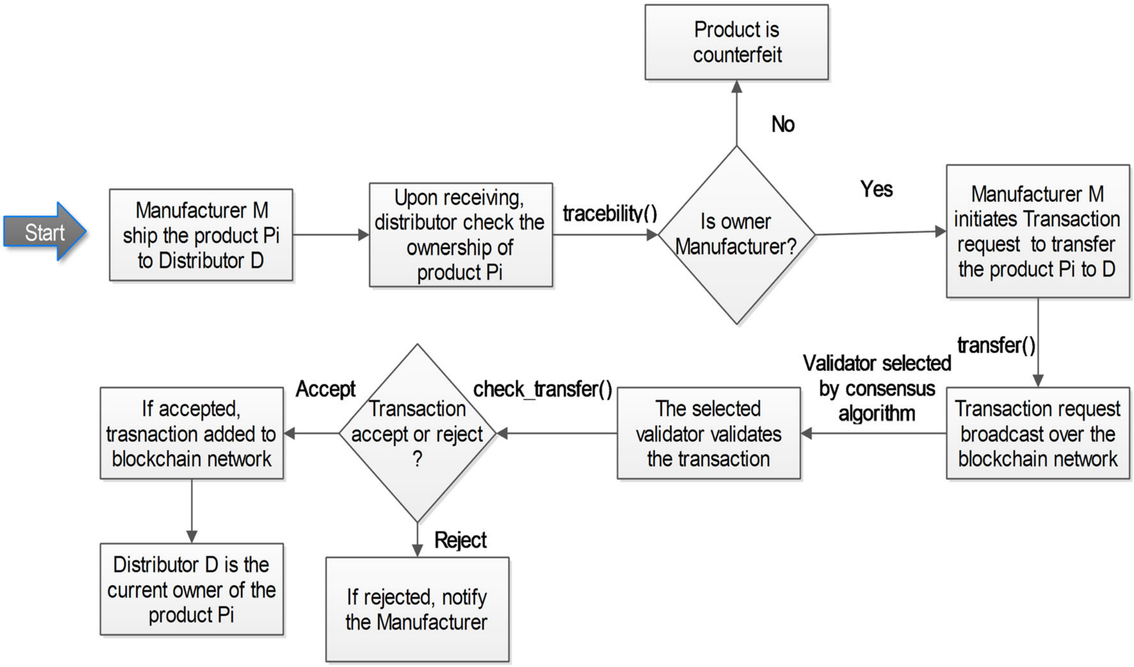
Start
Manufacturer M ship the product Pi to Distributor D
Upon receiving, distributor check the ownership of product Pi
tracebility()
Is owner Manufacturer?
No
Product is counterfeit
Yes
Manufacturer M initiates Transaction request to transfer the product Pi to D
transfer()
Transaction request broadcast over the blockchain network
Validator selected by consensus algorithm
The selected validator validates the transaction
check_transfer()
Transaction accept or reject ?
Accept
If accepted, trasnaction added to blockchain network
Distributor D is the current owner of the product Pi
Reject
If rejected, notify the Manufacturer

Fig. 8 Process of product transfer

5.4.2 Data validation

When a transaction is submitted, it is intended for two actions ‘create’ or ‘transfer’. If a validator receives the transaction with ‘create’ action, then the validator invokes Algorithm 4 (Check_create()) to match the signer’s key with hardcoded $M1_{PK}$. If match is found, then the transaction is accepted, otherwise it is rejected. If the transaction is requested with ’transfer’ action, then the validator invokes Algorithm 7 Check_Transfer() to match the current owner of the requested transaction with the Ts of product Pi. The transaction is accepted if match is found, otherwise it is rejected.

5.4.3 Data storage

If the validator accepts the transaction, then the transaction is stored. Because each node preserves the ledger copy, blockchain becomes more immutable; hence, blockchain consumes substantial storage space. In the view of the storage problem, only essential data are used to form a transaction.

5.4.4 Counterfeit protection

In this section, the possible scenarios related to the counterfeiting of a product and how the proposed system prevents the counterfeiting of the product are discussed. We refer to existing and new owners as seller and buyer, respectively, for convenience. In this case, three possible situations can occur.

(i) The seller attempts to sell a counterfeit product with a fake product code.
(ii) The seller attempts to sell a counterfeit product with a genuine product code, but the seller does not own the genuine product.
(iii) The seller owns a genuine product and counterfeit product with the same genuine product code.

In the first situation, if a seller tries to sell the counterfeit product, then the buyer denies purchasing the product because they can verify the product code by using Algorithm 8 and Algorithm 8 does not provide any details for the fake product code. In the second situation, the buyer queries Algorithm 8 about the genuine product code. Algorithm 8 provides the details of the current owner and origin. Because the seller does not own the product, they would not be the owner of the product. In this situation, the buyer can refuse to buy the product. In the third situation, when the seller can sell the counterfeit product (pretending product code Pi) and transfer product ownership to the buyer by calling the Algorithm 6 with Pi, the seller would lose ownership of the genuine product Pi, and under this condition, the seller would not be able to sell product Pi. Hence, no economic benefit is gained by selling the counterfeit product because the genuine product is costlier than the counterfeit product.

5.5 Security and privacy analysis

In this section, security and privacy of proposed system is evaluated.

5.5.1 Security analysis

The security of the proposed system is proved through the following theorems.

Theorem 1 *Suppose user* $U1$ *obtains the product creation algorithm* (*Algorithm* 1), which is meant for manufacturer M1. However, they can send the creation request but cannot create a transaction on blockchain.

Proof If a user gets Algorithm 1 and requests the transaction, then the validator will not validate the transaction because signer's key must match with M1's public key. A public key is created using the private key, and private key must be kept secret. □

Theorem 2 *Suppose a user can obtain all blockchain data and the public keys of all users, still he will not be able to view all data.*

Proof The public keys of all users are public to all users. A user is identified by their public key. All participants have a copy of blockchain data. The data is stored in the encrypted form, and SYM_KEYi is stored on the CDS. If the user is the owner of the product, they quickly can obtain SYM_KEYi by calling the REST API of the CDS with SHA256 hash (Pi_{H1}) of product Pi. If a user does not have the product, only a listed validator can obtain SYM_KEYi by calling the REST API of the CD with SHA244 hash and (Pi_{H2}) of product Pi and public key, and a validator can see the owner of the product but cannot see product Pi. □

Theorem 3 *Suppose a validator calls REST API with* V_{pub_key} *and* Pi_{H2} *to obtain SYM_KEYi of Pi. The public keys of all users are known. Adversaries can call REST API with* V_{pub_key} *and* Pi_{H2} *to obtain SYM_KEYi of Pi; however, they will not be able to acquire SYM_KEYi.*

Proof When a validator calls the REST API of the CDS with V_{pub_key} and Pi_{H2} to obtain SYM_KEYi of Pi, REST API acquires the SYM_KEYi of Pi_{H2} and encrypts SYM_KEYi with V_{pub_key} and returns it. Only the legitimate validator with V_{pri_key} can decrypt it. The V_{pri_key} is a secret key only known to the validator and not known to others. □

5.6 Privacy analysis

To encrypt the transaction (T_{Pi}) of product Pi, hash SHA244 and AES 128 bit symmetric key encryption are used. A blockchain node contains the transactions of N products. For each product Pi, a dedicated SYM_KEYi exists. Where $0 < i \leq N$

The transaction data of product Pi is given as follows.

$$T_{Pi} = (Pi_{H2},\ \text{action}, \text{Encrypted}_{\text{data}}, \text{timestamp}), \tag{1}$$

where

Pi_{H2} = SHA244(Pi)

Encrypted_data = AES (CO_{PK} || NO_{PK})

Equation 1 presents the transaction for product Pi.

Consider an illegitimate user became a part of the blockchain network; then, the user may obtain all the blockchain data. The first field SHA244(Pi) is in the hash form, which is irreversible; thus, Pi cannot be achieved. To encrypt $CO_{PK} \| NO_{PK}$, AES 128 bit encryption is used, and 1.02×100^{18} years are required to crack AES 128 bit encryption by using the brute force attack [41]. To obtain the SYM_KEYi of product Pi from the centralised server, user U has to send its public key (U_{PK}) in the URL, and if illegitimate user U sends the public key (UL_{PubK}) of legitimate user UL in URL, then the CDS returns the response as follows. SYM_KEYi_{en} = ECC_encrypt (UL_{PubK}, SYM_KEYi)

This SYM_KEYi_{en} key can only be decrypted using the private key of user UL. The proposed system uses a 256 bit ECC key. To break 256 bit ECC, 10^{28} MIPS years are required [42], and the current desktop computer can provide 2,356,230 MIPS [43]. Thus ECC 256 would take 4.24×10^{21} ($10^{28} \div 2,356,230$) years to break. Only the

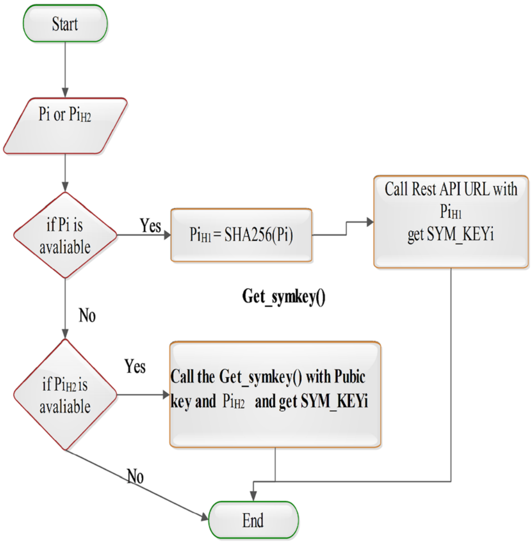
Start
Pi or Pi_{H2}
if Pi is avaliable
Yes
Pi_{H1} = SHA256(Pi)
Call Rest API URL with Pi_{H1} get SYM_KEYi
Get_symkey()
No
if Pi_{H2} is avaliable
Yes
Call the Get_symkey() with Pubic key and Pi_{H2} and get SYM_KEYi
No
End

Fig. 9 Process of accessing the symmetric key of product Pi.

legitimate validator and product owner can access the symmetric key. Figure 9 presents the process of accessing the symmetric key of Pi.

5.6.1 Rationale to choose ECC over RSA

ECC algorithm with 256 bits key provides the same security as provided by RSA with 2048 bits key. ECC consumes less computing power and battery resource. RSA is found to be very effective in encryption but slow in decryption while ECC is slow in encryption but very effective in decryption. ECC uses smaller keys, cyphertexts, and signatures [44]. Overall, the ECC is more secure and efficient than RSA [45].

6 Evaluation of the proposed system

Bai and Sarkis [46] reported the requirement of an assessment and evaluation model for blockchain technologies in the supply chain because no proposed models are available to evaluate the blockchain functionality; by contrast, all models suggest implementation and not the assessment of success. The current evaluative technologies only assess traditional models. Blockchain-based system performance depends on various factors such as network bandwidth [47], the blockchain platform used [12, 48], total number of nodes in blockchain, and machine configuration. Thus, the proposed system can not be compared with existing blockchain supply chain solution. To develop a large testbed for simulating and evaluating the performance of applications such as blockchain would require high effort [49]. Therefore, to form the proposed system, five machines are used, and among five, one machine is used for the CDS. Table 1 shows the parameters used to form the entire system.

The performance of the proposed system is evaluated. For performance evaluation, two blockchain systems were used; the first system is baseline blockchain without encryption algorithms (without proposed approach) and the second systems is baseline blockchain with the proposed approach. For performance evaluations, T_{TR}, the transaction used to transfer the product, is considered. To measure the overhead of the proposed system, T_{TR} is used as the transactions of the CDS for SYM_KEYi. First, stress is tested to determine how many simultaneous transactions the system can accept. The results of testing experiments indicated that the baseline blockchain system can simultaneously handle maximum 20 transactions. The submitter is built such that after submitting 20 transactions, it takes a 1-s pause to let the validator handle transactions. T_{TR} is the most expensive transaction in the proposed system because it includes encryption and decryption computing, calling the CDS, and updating product ownership. To assess blockchain performance, we submitted a different number of transactions and recorded transaction submission and committed time. The built-in Python logger was used to track events. Figure 10 presents comparison performance submission and the completion of the transactions of baseline blockchain.

Same sets of the transaction were submitted to the blockchain of the proposed system. Figure 11 shows the results of transaction submission and completion.

The results of baseline blockchain and the proposed system are compared based on throughput and latency.

Throughput comparison the throughput is defined as the speed of addition of transactions to blockchain. Based on the evaluation results illustrated in Figs. 10 and 11 the average throughput can be calculated as follows:

$$T_{avg\_tput} = \frac{1}{n}\sum_{i=1}^{n}\frac{N_i}{t_{ci}}, \tag{2}$$

where t_{ci} denotes the transaction committed time, N_i represents the number of transactions occurred for the ith trial, and n signifies the number of conducted trials. On the basis of Eq. 2 and the results (Figs. 10, 11), the baseline throughput and proposed system throughput were estimated to be 18.3356 and 17.5276 respectively. These two values revealed an additional overhead of 4.406% introduced by the proposed system. Privacy algorithms did not introduce any significant difference.

Latency latency is defined as the processing time, that is, the time difference between submission and completion of the transaction. Based on our assessment, the latency is defined as follows.

Table 1 Parameters used in the evaluation

Parameters	Value
Machine specification	Dell 7th generation, CPU : Intel Core i3, RAM: 8 GB
Switch	8 Ports 1 Gb unmanageable switch
Hyperledger Sawtooth client	Python Client
Centralised database	MySQL, Memcache
Operating system	Ubuntu 16.0
REST API	Flask web application framework

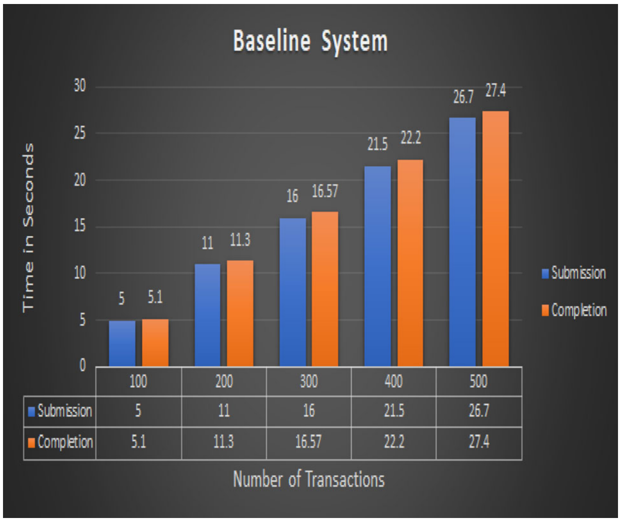

Fig. 10 Transaction without privacy

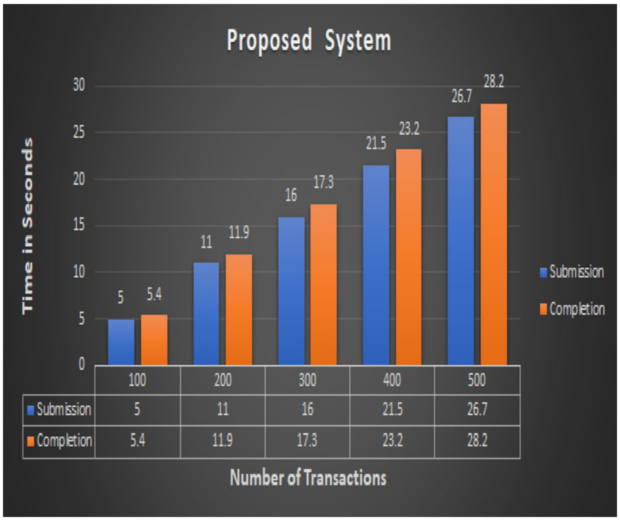

Fig. 11 Transaction of the proposed system

$$Lt = \frac{1}{n}\sum_{i=1}^{n}(t_{ci} - t_{si})\frac{1}{T_i}, \tag{3}$$

where t_{si} and t_{ci} denote the submission and completion times of transaction, respectively, of the ith trial. Ti represents the number of transactions submitted in the ith trial. Based on the evaluation baseline and proposed system, the latency of baseline blockchain and the proposed system were 0.00150 and, 0.00401 respectively. The latency of the proposed system and baseline is too low to compare. The combined result of baseline and proposed design are demonstrated in Fig. 12.

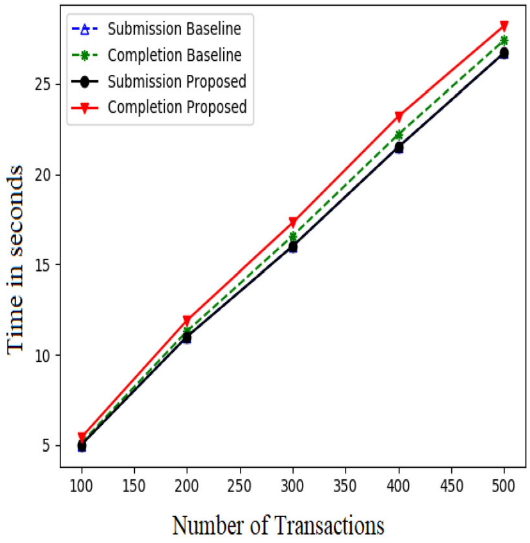

Fig. 12 Basline and proposed design

6.1 Time complexity analysis

The time complexity of the proposed system has been analysed for one complete cycle of product transfer transaction. The complexity of every step is as follows:

(i) *Product transfer* in product transfer, the user creates a request involving one symmetric key operation encryption step, denoted as α .

(ii) *Product validation* in product validation, the validator validates the transfer request based on rules. It involves one asymmetric key encryption and decryption step $(\beta + \gamma)$ and one symmetric key decryption step (δ).

Assuming all machines are of the same configuration, total complexity for one transfer cycle would be as follows:

$$TimeComplexity = O(\alpha + \beta + \gamma + \beta). \tag{4}$$

We accessed the proposed system's performance on a machine with an Corei7 7th generation CPU, 8 GB RAM and 1000 GB SSD. Table 2 provides the average time (in seconds) of α, β, γ, and δ for different number of operations.

Figure 13 demonstrates the visual representation of Table 2. The ECC decryption process takes more time than the ECC encryption does, and AES encryption and decryption process took much less compared to ECC.

6.2 Comparison of current platforms

To compare the proposed work, we selected various blockchain-based supply chain solutions. The comparative outcome is provided in Table 3. Most of the papers used Ethereum and Bitcoin public blockchains in their supply

Table 2 Time required for encryption and decryption process

Operations	γ	β	α	δ
1000	0.599176598	0.283522129	0.017428637	0.017507553
5000	3.11029911	1.489209414	0.077558517	0.0757792
10,000	6.225434303	3.001870155	0.158222437	0.160424948
15,000	9.281481028	4.501509666	0.225598097	0.25079751
20,000	12.3824501	6.126716375	0.311615705	0.37565136

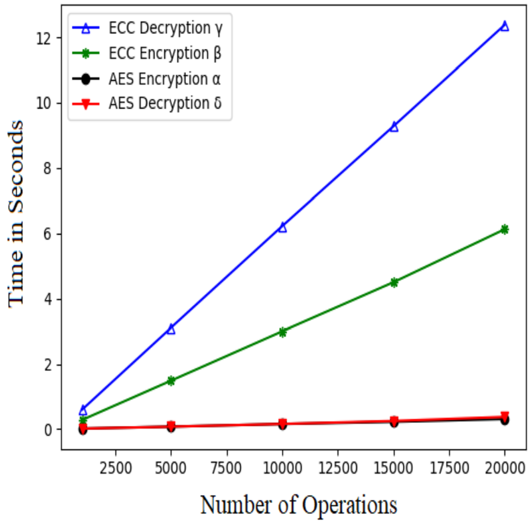

Fig. 13 Time complexity of encryption–decryption processes

chain innovation. However, due to the low efficiency of a public blockchain, it is not a good option in a supply chain. We use the Hyperledger Sawtooth consortium blockchain. The blockchain of a consortium is much safer than a public one. Furthermore, the privacy module on a consortium blockchain is better to build.

7 Conclusions

In this paper, the blockchain-based approach for the supply chain is proposed to address the problem of counterfeit products, privacy, and traceability. The proposed system allows to create, transfer, and track assets. The performance results of a proof of concept implementation obtained using Hyperledger Sawtooth, demonstrated that the additional overhead caused by the proposed system is minimal. The proposed system can support business models that require less than 17 tps. The proposed system can run on low configuration economical computer machines. The PoET consensus algorithm offered by Hyperledger Sawtooth is highly energy conserving. Privacy a critical concern because neither businesses nor customers want their information to be released on a public database. In the proposed system, permission blockchain with five nodes that allows users to store the information in encrypted form is implemented. The information is encrypted using the symmetric key, which is stored in the CDS. The CDS is designed such that it only returns the symmetric to a legitimate user. The limitation of proposed work is that it can handle only 17 tps; thus, it can be applied where the transaction rate is less than 17 tps. Another limitation is that validators depend on the CDS to obtain the symmetric key, and they must call CDS every time. Thus, the availability of network bandwidth of the CDS can highly

Table 3 Current platform comparison

	POMS [10]	B_verify [15]	Smart supply [13]	SCF [7]	Modum [9]	Our platform
Blockchain choice	Ethereum	Bitcoin	Ethereum	Hyperledger Fabric	Ethereum	Hyperledger Sawtooth
Transaction privacy	No	No	No	Yes	No	Yes
Consensus	PoW	PoW	PoW/BFT	BFT	PoW	PoET
System cost	High	High	Low	Low	High	Very low
Delay of transactions confirming on the blockchain	15 s	1 h	15 s	5 s	15 s	1 s
Tamper-resistant	Yes	Yes	Yes	Yes	Yes	Yes

influence the transaction committed time. A low network bandwidth can increase the transaction committed time. Multiple replicas of the CDS can alleviate the network bandwidth problem.

Acknowledgements I want to thank my supervisor Dr. Sanmeet Kaur and Maninder Singh, for their valuable efforts and suggestions.

Funding Not applicable.

Data availability Not applicable.

Code availability Hyperledger Sawtooth 1.01 is used with custom smart contract code.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Marchese, K.: Supply chain leadership. https://www2.deloitte.com/us/en/pages/operations-/articles/supply-chain-leadership.html. Accessed 29 July 2020
2. May, A.: No Blockchain to tackle supply chain failures exposed by COVID-19 and boost economic recovery (2020). https://www.weforum.org/press/2020/04/blockchain-to-tackle-supply-chain-failures-exposed-by-covid-19-and-boost-economic-recovery/. Accessed 29 July 2020
3. Stevekugel. 5 Reasons for Failure in Australian Small Business. https://insolvencyexperts.com.au/5-reasons-failure-australian-small-business/. Accessed 29 July 2020
4. Landi, H.: IBM rolls out blockchain network to address supply-chain issues caused by COVID-19 (2020). https://www.fiercehealthcare.com/tech/ibm-rolls-out-blockchain-network-to-match-healthcare-organizations-non-traditional-suppliers. Accessed 29 July 2020
5. O'Byrne, R.: 7 Reasons Why the Supply Chain Matters to Business Success. https://www.logisticsbureau.com/7-reasons-why-the-supply-chain-matters-to-business-success/. Accessed 29 July 2020
6. O'Byrne, R.: Blockchain technology is set to transform the supply chain (2019). https://www.logisticsbureau.com/h:ow-blockchain-can-transform-the-supply-chain/. Accessed 29 July 2020
7. Du, M., Chen, Q., Xiao, J., Yang, H., Ma, X.: Supply chain finance innovation using blockchain. IEEE Trans. Eng. Manag. **67**(4), 1045–1058 (2020)
8. Biswas, K., Muthukkumarasamy, V., Tan, W.L.: Blockchain based wine supply chain traceability system. In: Future Technologies Conference (FTC), 2017
9. Bocek, T., Rodrigues, B.B., Strasser, T., Stiller, B.: Blockchains everywhere—a use-case of blockchains in the pharma supply-chain. In: Proceedings of the IM 2017—2017 IFIP/IEEE International Symposium on Integrated Network and Service Management, pp. 772–777 (2017)
10. Toyoda, K., Mathiopoulos, P.T., Sasase, I., Ohtsuki, T.: A novel blockchain-based product ownership management system (POMS) for anti-counterfeits in the post supply chain. IEEE Access **5**, 17465–17477 (2017)
11. Malik, S., Kanhere, S.S., Jurdak, R.: ProductChain: scalable blockchain framework to support provenance in supply chains. In: NCA 2018—2018 IEEE 17th International Symposium on Network Computing and Applications (2018)
12. Caro, M.P., Ali, M.S., Vecchio, M., Giaffreda, R.: Blockchain-based traceability in agri-food supply chain management: a practical implementation. In: IoT Vertical and Topical Summit on Agriculture—Tuscany. IOT Tuscany, vol 2018, 2018 (2018)
13. Su, S., Wang, K., Kim, H.S.: Smartsupply: smart contract based validation for supply chain blockchain. In: Proceedings—IEEE 2018 International Congress on Cybermatics: 2018 IEEE Conferences on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, iThings/Green, 2018
14. Leng, K., Bi, Y., Jing, L., Fu, H.C., Van Nieuwenhuyse, I.: Research on agricultural supply chain system with double chain architecture based on blockchain technology. Future Gener. Comput. Syst. **86**, 641–649 (2018)
15. Chod, J., Trichakis, N., Tsoukalas, G., Aspegren, H., Weber, M.: On the financing benefits of supply chain transparency and blockchain adoption. Manag. Sci. (2019). https://doi.org/10.1287/mnsc.2019.3434
16. Malik, S., Dedeoglu, V., Kanhere, S.S., Jurdak, R.: TrustChain: trust management in blockchain and IoT supported supply chains. In: Proceedings—2019 2nd IEEE International Conference on Blockchain, Blockchain 2019 (2019)
17. Lin, Q., Wang, H., Pei, X., Wang, J.: Food safety traceability system based on blockchain and EPCIS. IEEE Access (2019). https://doi.org/10.1109/ACCESS.2019.2897792
18. Shahid, A., Almogren, A., Javaid, N., Ahmad Al-Zahrani, F., Zuair, M., Alam, M.: Blockchain-based agri-food supply chain: a complete solution. IEEE Access (2020). https://doi.org/10.1109/ACCESS.2020.2986257
19. Liu, Z., Li, Z.: A blockchain-based framework of cross-border e-commerce supply chain. Int. J. Inf. Manag. (2020). https://doi.org/10.1016/j.ijinfomgt.2019.102059
20. Iqbal, R., Butt, T.A.: Safe farming as a service of blockchain-based supply chain management for improved transparency. Clust. Comput. **23**(3), 2139–2150 (2020)
21. Amir Latif, R.M., Farhan, M., Rizwan, O., Hussain, M., Jabbar, S., Khalid, S.: Retail level blockchain transformation for product supply chain using truffle development platform. Clust. Comput. (2021). https://doi.org/10.1007/s10586-020-03165-4
22. Kshetri, N., Loukoianova, E.: Blockchain adoption in supply chain networks in Asia. IT Prof. (2019). https://doi.org/10.1109/MITP.2018.2881307
23. Everledger Diamond Platform. https://diamonds.everledger.io. Accessed 29 July 2020
24. Analytics in Insurance Fraud: The Fight Before the Claim. https://www.insurancenexus.com/fraud/analytics-insurance-fraud-fight-claim. Accessed 29 July 2020
25. Blockchain Based Anti-counterfeit Solution. Accessed 29 July 2020
26. Karunamoorthi, K.: The counterfeit anti-malarial is a crime against humanity: a systematic review of the scientific evidence. Malar. J. **13**(1), 1–13 (2014)
27. Verisart. https://verisart.com/. Accessed 29 July 2020
28. Rajnic, M., Kacina, J., Harler, M.C.: SophiaTX Whitepaper: The Blockchain for Business (2017). Accessed 29 July 2020
29. Every product has a story. https://www.provenance.org. Accessed 29 July 2020
30. Chronicled. https://www.chronicled.com/. Accessed 29 July 2020
31. Pierro, M.D.I.: What Is the blockchain? Comput. Sci. Eng. **19**(5), 92–95 (2017)
32. Siva Sankar, L., Sindhu, M., Sethumadhavan, M.: Survey of consensus protocols on blockchain applications. In: 2017 4th

International Conference on Advanced Computing and Communication Systems, ICACCS 2017 (2017)

33. Zhang, S., Lee, J.H.: Analysis of the main consensus protocols of blockchain. ICT Express (2020). https://doi.org/10.1016/j.icte.2019.08.001
34. Eyal, I.: Blockchain technology: transforming libertarian cryptocurrency dreams to finance and banking realities. Computer (2017). https://doi.org/10.1109/MC.2017.3571042
35. Castro, M., Liskov, B.: Practical Byzantine fault tolerance. In: Proceedings of the Symposium on Operating System Design and Implementation, 1999
36. Feng, Q., He, D., Zeadally, S., Khan, M.K., Kumar, N.: A survey on privacy protection in blockchain system. J. Netw. Comput. Appl. **126**(May 2018), 45–58 (2019)
37. Amarasinghe, N., Boyen, X., McKague, M.: A survey of anonymity of cryptocurrencies. In: ACM International Conference Proceeding Series, 2019
38. Wang, R., Ye, K., Meng, T., Xu, C.Z.: Performance evaluation on blockchain systems: a case study on Ethereum, Fabric, Sawtooth and Fisco-Bcos. In: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (2020)
39. Ampel, B., Patton, M., Chen, H.: Performance modeling of Hyperledger Sawtooth blockchain. In: 2019 IEEE International Conference on Intelligence and Security Informatics, ISI 2019 (2019)
40. Moschou, K., Theodouli, A., Terzi, S., Votis, K., Tzovaras, D., Karamitros, D., Diamantopoulos, S.: Performance evaluation of different Hyperledger Sawtooth transaction processors for Blockchain log storage with varying workloads. In: Proceedings—2020 IEEE International Conference on Blockchain, Blockchain 2020 (2020)
41. Arora, M.: How secure is AES against brute force attacks? https://www.eetimes.com/document.asp?doc_id=1279619#. Accessed 29 July 2020
42. Cai, T., Chen, W., Yu, Y.: BCSolid: a blockchain-based decentralized data storage and authentication scheme for solid. In: Communications in Computer and Information Science. Springer, Singapore (2020)
43. Instructions per second (2020). https://en.wikipedia.org/wiki/Instructions_per_second. Accessed 29 July 2020
44. Mahto, D., Yadav, D.K.: RSA and ECC: a comparative analysis. Int. J. Appl. Eng. Res. **12**(19), 9053–9061 (2017)
45. Maletsky, K.: RSA vs ECC Comparison for Embedded Systems. Atmel-8951A-CryptoAuth-RSA-ECC-Comparison-Embedded-Systems-WhitePaper_072015 (2015)
46. Bai, C., Sarkis, J.: A supply chain transparency and sustainability technology appraisal model for blockchain technology. Int. J. Prod. Res. **58**(7), 2142–2162 (2020)
47. Zhang, P., White, J., Schmidt, D.C., Lenz, G., Trent Rosenbloom, S.: FHIRChain: applying blockchain to securely and scalably share clinical data. Comput. Struct. Biotechnol. J. **16**, 267–278 (2018)
48. Moubarak, J., Filiol, E., Chamoun, M.: On blockchain security and relevant attacks. 2018 IEEE Middle East and North Africa Communications Conference, MENACOMM 2018, pp. 1–6 (2018)
49. Alzahrani, N., Bulusu, N.: A new product anti-counterfeiting blockchain using a truly decentralized dynamic consensus protocol. Concurr. Comput. Pract. Exp. (2020). https://doi.org/10.1002/cpe.5232

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Mohit Mohit received his Bachelor's Degree in Computer Engineering from UIET Kurukshetra University, Kurukshetra in 2009, and holds a Master's Degree in Computer Science from Thapar Institute of Engineering and Technology. He has worked in IBM, Teramatrix, and Sapient as a Developer. He is a Certified Ethical Hacker (C|EH), and Security Analyst (ECSA) from EC-Council USA. He is the author of books Python Penetration Testing Essentials, Learn Python in 7 days (by Packt) and Python for Developer (by BPB). He is the active Journal Reviewer of a SCI journal. He is pursuing a Doctoral Degree in Computer Science and Engineering Department from Thapar Institute of Engineering and Technology.

Sanmeet Kaur received her Bachelor's Degree from Guru Nanak Dev University, Amritsar in 2001, and holds a Master's Degree in Software Engineering from Thapar Institute of Engineering and Technology, as well as a Doctoral Degree specialization in Network Security from Thapar University. She is currently working as an Associate Professor in Computer Science and Engineering Department at Thapar University. Dr. Kaur is on the Roll-of-Honor at EC-Council USA, being certified as Ethical Hacker (C|EH).

Maninder Singh received his Bachelor's Degree from Pune University in 1994, and holds a Master's Degree, with Honors in Software Engineering from Thapar Institute of Engineering and Technology, as well as a Doctoral Degree specialization in Network Security from Thapar University. He is currently working as Professor in Computer Science and Engineering Department at Thapar University. Dr. Singh is on the Roll-of-Honor at EC-Council USA, being certified as Ethical Hacker (C|EH) Security Analyst (ECSA), and Licensed Penetration Tester (LPT). Dr. Singh has completed many consultancy projects for the renowned national bank(s).