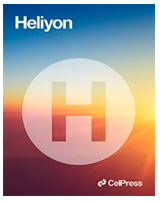
Heliyon 10 (2024) e40273

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon



Research article

Blockchain based solution for secure information sharing in pharma supply chain management☆

Adla Padma, Mangayarkarasi Ramaiah *

School of Computer Science Engineering and Information systems, Vellore Institute of Technology, Vellore, India

ARTICLE INFO

Keywords:
Ethereum
IoT
Pharma supply chain management
Secured payment
Smart contracts

ABSTRACT

Pharmaceutical supply chain management (PSCM) aims to alleviate logistical challenges. However, traditional online pharma systems face issues during implementation, particularly regarding transparency and fostering mutual trust among stakeholders. The primary security goals for a supply chain management (SCM) solution are ensuring authentication, confidentiality, data provenance, and auditability. The proposed blockchain-based solution (BPSCM) is implemented in three phases: registration, pharmaceutical product circulation, and secure payment. The registration phase computes the identification number upon the hashed private key along with the Edwards-curve digital signature algorithm (EdDSA) for all the stakeholders. The pharm product circulation phase implements the transactions among the participants by developing smart contracts where cryptographic operators ensure data provenance. The security analysis demonstrates that the framework effectively mitigates impersonation and collusion attacks. Performance metrics, including gas consumption, throughput, latency, and computational cost, were examined and compared to standard PSCM frameworks to evaluate the BPSCM's effectiveness.

1. Introduction

PSCM is a complex network of companies, people, activities, technology, information, and resources that manufacture and distribute pharmaceutical goods. Supply chain logistics entails many responsibilities, including planning, sourcing, production, delivery, and quality assurance. As the healthcare sector transitions towards custom-made items managed on an order-by-order basis, the role of statistical data mining in enhancing supply chain activities, strategic planning, design, execution, and personnel management becomes increasingly crucial [1,2]. According to the Health Research Funding Organization, 30 % of pharmaceuticals sold in underdeveloped nations are counterfeit. According to a recent World Health Organization (WHO) study, children are the most frequent victims of counterfeit medications, which are one of the main causes of death in developing nations. Counterfeit drugs not only endanger human lives but also incur significant financial losses for the pharmaceutical industry [3]. Fake medicines have played a vital role in public health and patient safety for the last few years. There is a substantial need to develop programs to prevent and detect fake medicines, including falsely labeled, unapproved, falsified, and counterfeit [50].

Since the 1980s, stakeholder roles have significantly impacted the supply chain. In the past, stakeholders had a passive role, with goods being promoted by upstream suppliers regardless of the retailer's understanding of customer demand. However, this has

☆ This work was supported by the Vellore Institute of Technology, Vellore, Tamil Nadu, India.

* Corresponding author.
E-mail addresses: adla.padma2020@vitstudent.ac.in (A. Padma), rmangayarkarasi@vit.ac.in (M. Ramaiah).

https://doi.org/10.1016/j.heliyon.2024.e40273
Received 4 May 2024; Received in revised form 1 November 2024; Accepted 7 November 2024
Available online 8 November 2024

changed due to rapid technological advancements, intense market competition, and sophisticated supplier networks. Stakeholders now play a crucial role in the supply chain, particularly in predicting real-time customer demand. This shift has influenced retailers' changing positions, leading to the outsourcing of non-core activities, reduction of excess inventory, and increased reliance on information technology [4].

The pharmaceutical sector is critical to protecting public health by providing life-saving pharmaceuticals to individuals worldwide. Establishing an effective and dependable supply chain is crucial. However, the present PSCM facing various issues in detecting counterfeit products, a lack of transparency, ineffective monitoring, and security flaws. These difficulties could hinder access to critical pharmaceuticals, jeopardize patient safety, and raise overall costs throughout the system. The existence of counterfeit drugs in the PSCM is a significant threat to the community's health. Creating an effective traceability system is critical for verifying the legitimacy of medications throughout their entire existence. This implies tracing the origin of materials, processing, packaging, transportation, and pharmacy delivery. In addition to counterfeiting, the pharmaceutical industry faces several significant challenges, such as the inability to verify the provenance of medicines during pandemics. The issue of uncontrolled drug prices needs to be addressed. Foreign-imported Active Pharmaceutical Ingredients (APIs) can disrupt production and mislead consumers [55]. Gupta and Namasudra [64] have presented a framework to accelerate virtual machine migration in cloud computing by reducing downtime, migration time, and data transfer rate. However, the system's performance was affected by complex master-slave coordination.

Recent technological advancements have opened up promising avenues to address these challenges. The shift from centralized cloud platforms to decentralized and distributed technology is a significant development. Blockchain offers a groundbreaking approach to supply chain tracing, utilizing a chain of blocks in chronological order managed by network participants. Cryptographic hash techniques ensure these blocks' immutability, serving as a transaction history record. With its unique features of immutability, transparency, and security, Blockchain Technology (BT) has emerged as a disruptive force in various sectors, including pharmaceuticals [52,53]. Researchers and industry experts are exploring the potential of blockchain in the PSCM, keen to leverage its benefits to overcome key challenges and create a more sustainable, efficient, and secure environment. BT in electronic SCM allows for real-time transaction recording and tracking, ensuring dependability and preventing counterfeiting.

Supply chain analysis entails assessing each economic agent's function and contribution across the supply chain, from raw material procurement to final product delivery. Developing countries require assistance in managing supply chains, which include political instability, corruption, and a restricted adoption of modern technology [6]. Information sharing across supply chain trading partners is becoming progressively crucial as the economy grows unstable. Early prediction of end-user demands can reduce future uncertainty and eliminate leftover stock. Furthermore, supply chain visibility enables managers to react efficiently to timely data and facilitates efficient planning [7]. Moreover, there is a pressing need for equitable and sustainable drug distribution, considering disease burden, availability, accessibility, and environmental sustainability. In this context, enhancing inventory management processes, implementing targeted interventions, and harnessing technologies like blockchain can significantly enhance the performance of the PSCM and improve healthcare delivery outcomes [5]. Table 1 provides a comprehensive overview of the involvement of various stakeholders in SCM, both with and without the integration of blockchain.

To comply with regulations pertaining to the European Union General Data Protection Regulation (EUGDPR), PSCM systems must adhere to various fundamental guidelines, including consent-based data processing, which ensures that personal data is only gathered and handled with the approval of stakeholders and customers. The system must also ensure transparency and accountability by giving explicit information regarding data management. Data integrity and accuracy are critical, which means that data must be maintained correctly and securely, with safeguards against unauthorized access or deletion. Furthermore, privacy by design should be built into the system from the start, ensuring that data security is a priority throughout the development process. Data access should be limited to authorized entities, and all personal data activity must be documented. Finally, personal data should be erased once it is no longer needed. Regulatory-compliant designs ensure hashed transaction data (e.g., drug batches, expiration dates) is stored on the blockchain while sensitive data remains within secure systems to comply with regulations like GDPR.

The compatibility aspect of the proposed model with existing PSCM systems is essential to ensure seamless adoption and practical implementation. BPSCM enhances data transparency, security, and efficiency within the network without requiring a complete infrastructure overhaul. The developed system can integrate with traditional PSCM systems through APIs (REST), middleware (data integration, service-oriented), and regulatory-compliant designs. APIs allow real-time data exchange between drug tracking and order management, ensuring legacy system data is incorporated into the blockchain for provenance and transparency. Middleware translates data from traditional PSCM systems into a blockchain-compatible format.

Shankar et al. [56] presented a multi-signature schema for digital documents using the EdDSA algorithm. Signing and verification are performed using generated public and private keys. It is cost-effective and secure, and the computational cost is less than other

Table 1
Pharma supply chain using BT.

PSCM role players	PSCM without blockchain	Merits of blockchain
Suppliers (P_s)	Drug raw molicules	Storage of drug raw molicules
Retailers (P_r)	Traceability issues	Achieves traceability
Drug manufactures (P_{dm})	Drug production	Storage of drugs
Distributors (P_d)	Lack of tracking	Decentralized tracking
IoT container (P_{IoT})	Lack of tracking	Decentralized tracking using IoT sensors
Consumers (P_c)	Drug quality issues	Attains compelled quality drugs

cryptography techniques. Datta and Namasudra [59] introduce a novel blockchain-based Electronic Medical Record (EMR) sharing framework that leverages Mobile Edge Computing (MEC) and consumer electronic devices. This framework incorporates additional security layers using Advanced Encryption Standards (AES) encryption, stores encrypted EMRs and diagnosis reports on InterPlanetary File System (IPFS) and utilizes smart contracts and the Proof of Authority (PoA) consensus algorithm for efficient management and faster transactions. However, this schema needs more comprehensive detail in terms of the data input process and the potential for malicious entities to transmit irrelevant data.

Lou et al. [8] have presented a blockchain-based Supply Chain Electronic System Cooperation Framework (SESCF) to address authentication challenges and delays in capital flow, logistics, and product circulation. Radio Frequency Identification (RFID) is used for product identification to ensure the integrity of the SESCF model. Hasan et al. [49] have developed a blockchain-based model to handle challenges related to shipment by integrating IoT. Data is collected through IoT sensors and stored in blockchain-enabled databases in the cloud. Smart contracts are implemented to monitor deviation while collecting the data from sensors to implement secure payment, authentication, and product traceability. However, this model's transaction processing time is high.

Datta and Namasudra [62] have presented a blockchain-based counterfeit drug schema by adopting an IPFS server for off-chain storage and Proof of Work (PoW) for consensus verification and block validation. However, this schema fails to address the smart contract vulnerabilities. Mackey and Nayyar [50] have proposed smart contract-based models for counterfeit drugs. However, finding breaches after a customer completes payment can lead to a time-consuming process to verify and get a refund, posing a significant risk to the customer. These lead to a direct impact on customers for poor quality medicine delivery and business growth. In this case, the temperature of the containers might be crucial to prevent quality impairment and shipping contamination. Dwivedi et al. [9] have presented a blockchain-based PSCM system with reliable information sharing. The consensus process verify the authenticity of each new block and transaction, while the smart contract was employed to ensure secure cryptographic key distribution. However, intruders may compromise the security aspects of the included Certificate Authority (CA), which may again encourage external and internal adversaries to execute data tampering and collusion attacks.

Motivated by the vital significance of a realistic supply chain solution and the widespread adoption of blockchain and IoT [10] technology across numerous industries, we propose a BPSCM to connect various stakeholders in a trusted environment and promote product sharing. Despite technical hindrances, the candidate model is designed to significantly improve trust, efficiency, coordination, and transparency, providing a clear view of the entire supply chain process. The IoT sensors are implemented in the containers to facilitate product traceability and ensure product quality. They detect and report temperature deviations in the smart contract, notifying all stakeholders of the potential impact on the integrity of medical products. BT is used to accomplish the security features of secrecy and authenticity, achieved using symmetric and EdDSA algorithms. Smart contracts regulate interactions between multiple stakeholders to efficiently guarantee high-end customer satisfaction without involving intermediates. An energy-efficient Proof of Stake (PoS) consensus protocol was utilized for transaction validation and block mining process. Another challenge associated with the PSCM is ensuring product integrity. To implement this, the proposed BPSCM smart contract has been designed in such a way as to continuously monitor the product quality throughout the life cycle and raise alerts to appropriate stakeholders about the deviation incurred in maintaining temperature to prevent product contamination. The major contributions of this work are as follows.

1. An Ethereum smart contract-based model is proposed to ensure data provenance, transparency and integrity of the BPSCM framework.
2. Decentralized storage IPFS and EdDSA are used to store transactions and achieve confidentiality, scalability, high transaction throughput, and reduced computational cost.
3. Phases of the presented framework registration, authentication, and secure payment were implemented through deterministic Smart contracts. PoS consensus has been used to accelerate block mining and transaction validation.
4. The proposed smart contract was analyzed against security vulnerabilities using Oyente and the MyThril tool. The performance analysis of BPSCM performs better than the existing models.

The remainder of this paper is organized as follows: Section 2 reviews existing models' key contributions and limitations. Section 3 provides background on our proposed model, including entities and security goals. Section 4 discusses the proposed BPSCM and its corresponding smart contract algorithms. Section 5 presents the testing and validation, and security analysis. Section 6 shows the performance analysis of the proposed model. Finally, Section 7 concludes the paper.

2. Related works

The automated supply chain faces technical hindrances in ensuring consistency in maintaining the product's state among the diverse stakeholders. Blockchain is an efficient tool that facilitates transparent product state implementation. Preventing fake product entry for the candidate BPSCM is vital. Hence, this section analyzes the various blockchain-enabled supply chain frameworks that mitigate the breaches encountered through counterfeit products and other challenges. Mishra et al. [11] introduced an efficient blockchain-based framework that utilizes smart contracts. This innovative approach ensures secure and transparent tracking of pharmaceutical records. The framework chunks the data for efficient search to implement parallel searching. However, this model is appropriate for small-scale networks, even if it performs best.

Sharma et al. [63] proposed a secure cloud storage system integrating BT. This schema utilizes smart contracts, encryption, and optimization techniques to identify errors and code regeneration. However, the computational overhead of blockchain operations affected the system's performance. Namasudra and Sharma [65] proposed a blockchain-based secure cab-sharing system. This schema

employs re-encryption, delegated proof of stake consensus, and a reputation system to ensure security and privacy. However, the system fails to address the authentication services.

Yazdinejad et al. [66] have presented an energy-efficient software-defined networking (SDN) architecture for IoT networks using blockchain. This framework used a cluster structure with a new routing protocol for secure access control. However, the system performance is higher throughput, lower delay, and lower energy consumption compared to traditional methods. Yazdinejad et al. [67] present a novel anomaly detection framework for blockchain-based IIoT networks in smart factories. This schema utilizes a cluster-based architecture, federated learning, and various machine learning models to improve efficiency and detect anomalies. Additionally, the effectiveness of the anomaly detection models may vary depending on the specific characteristics of the IIoT network. Yazdinejad et al. [68] proposed a secure, intelligent fuzzy blockchain framework for network attack detection in IoT environments. The framework incorporates a fuzzy deep learning model, fuzzy control systems, metaheuristic algorithms for optimization, and fuzzy matching for fraud detection. Evaluation results demonstrate its threat detection and decision-making effectiveness within blockchain-based IoT networks.

Yazdinejad et al. [69] have presented an Auditable Privacy-Preserving Federated Learning (AP2FL) framework for medical devices. AP2FL uses a trusted execution environment to prevent data leakage during training and aggregation. Active Personalized Federated Learning (ActPerFL) and Batch Normalization (BN) were used to aggregate user updates and identify data similarities for non-IID data. Yazdinejad et al. [70] have proposed a hybrid privacy-preserving federated learning framework tailored for Next-Generation Internet of Things (NG-IoT) environments. This framework employed two-trapdoor homomorphic encryption and server protocol to mitigate the impact of irregular users, while the asynchronous hybrid algorithm reduced user dropout rates. Performance evaluations demonstrated the framework's superiority over existing solutions in terms of functionality, accuracy, and reduced system overheads.

One of the most critical issues facing the pharmaceutical sector is the detection of counterfeit products, which poses substantial risks to consumers and the environment. Jha et al. [12] introduced a Hyperledger Fabric (HLF) based blockchain solution in this context. This groundbreaking framework, consisting of ledger, smart contract, and user interface modules, holds immense potential to bolster the safety of pharmaceutical products by effectively identifying counterfeit drugs. However, it is crucial to note that this solution is primarily designed to track falsified medication supplied within authorized supply networks. Al-Farsi et al. [13] focused on addressing the internal and external attacks on the entities involved in SCM. It provides transparent interaction between participant entities by allowing business rules. These dynamic rules protect the system from internal and external attacks. However, it handles safe and unsafe constraints instead of valid constraints.

Ma et al. [14] proposed vulnerability detection on transactions using a hierarchical graph attention network. They utilized the Abstract Syntax Tree (AST) and control flow graph to analyze the smart contract functions. Since the flow of the transactions is precisely recorded through the AST and control flow graph, the model can detect vulnerabilities more quickly and precisely than the other detection models. Subramanian et al. [15] proposed that the NEM blockchain alleviates the burden of tracking drugs using mobile applications. The presented model uses NEM cryptocurrency and QR codes to label pharmaceutical products. Physicians, patients, and pharmacists have implemented product authenticity. Proof of Import (PoI) is used in smart contracts to validate the products. However, it does not include a direct drug monitoring system for doctors and patients. Similarly, Konapure and Nawale [16] developed a blockchain-based solution to enhance the traceability of pharmaceutical products through supply chain applications. Smart contracts are intended to reduce third-party involvement requirements.

In the traditional pharmaceutical supply chain, blind parties lead to information fragmentation, diminished participant responsibility, incomplete information at every level, and the potential for introducing counterfeit medications. These could endanger patients and result in financial loss. Fragmentation can lead to inaccurate demand estimates, which could harm consumers and even have fatal effects. To eliminate blind parties and data fragmentation between involved parties, Bapatla et al. [17] proposed an efficient smart contract-enabled PharmaChain model. Smart contracts and Proof of Concept (PoC) address access control policies and scalability issues. Regarding resilience against cyber-attacks, re-entrance attacks, and randomness by using the Oracle model.

Uddine [18] proposed a Medledger framework to identify phony medicine in the pharmaceutical ecosystem. The framework uses the HLF to ensure stakeholders' validation, authentication, and verification. Chain code exchanges critical information between participating entities over fine-grained access control for drug traceability. The Medledger network validates and tracks the activity of stakeholders using the smart contract modules drug registration, consignment accumulation, and transaction update contracts. Agrawal et al. [19] implement forward and backward supply chain models to minimize the delay and cost incurred by the drug delivery process. Utilizing Hyperledger Composer while modeling the assets and maintaining the transaction history greatly enhances security, transparency, and efficiency.

Similarly, Musamih et al. [20] proposed a blockchain-based solution by leveraging the Ethereum platform's smart contracts and encryption techniques to improve healthcare management's accuracy and transparency. This schema ensures authenticity by offering a combination of identity management and membership services. Streamlines processes and gives transparent, secure drug monitoring to various stakeholders, including manufacturers and patients. This innovative method has the potential to transform medication traceability and eventually improve security and patient safety across the entire healthcare supply chain, even though there are obstacles such as regulation and scalability. Chiacchio et al. [21] introduced a solution using Non-Fungible Tokens (NFTs) to enhance PSCM tracking and tracing capabilities. NFTs create a unique identifier for each parcel assembled by the packaging lines of a pharmaceutical factory. The NFTs are characterized by the Owner's Public Address to the manufacturer, making it impossible for others to append information to the smart contract. By minting NFTs, the system inherits the benefits of BT, offering enhanced transparency and security. This framework deployed in the VeChain Thor blockchain test network using PoA to guarantee enhanced scalability.

To mitigate the problems caused by counterfeit pharmaceuticals, Abbas et al. [22] presented a blockchain model to implement transparency and drug recommendation systems for consumers. The blockchain module tracks drug movement through the supply

chain, ensuring authenticity and transparency. The machine learning module uses reviews to recommend suitable medications to consumers. This combination is facilitated through HLF and REST API between both models to enhance the accuracy and efficiency of text classification models. Gaur et al. [23] proposed a blockchain and machine learning-based framework for effective data processing and storage. The model improves security and reduces storage consumption through Smart contracts. A support vector machine detects

Table 2
Review of traditional supply chain management using smart contracts.

Ref	Key observation	Domain	Implemented platform	Limitation	Metrics
[28]	Proposed framework utilizes a layered approach built on disruptive technologies to streamline logistics within the agricultural sector. Automation encompasses tasks within individual companies (intra) and between companies (inter). The proposal aims to minimize gas fees through a creative combination of different approaches.	Agriculture	Ganache, Truffle suite, Solidity	Test cases regarding consumer refunds and prompt delivery are yet to be considered.	Gas cost
[29]	The candidate framework mitigates food quality and payment issues through QR and bar codes.	Food SCM	DApp, HLF	An implementation needs more resources, which in turn leads to high costs.	Transaction time
[30]	BLS signature ensures authentication. Data Provenance details can be accessed without accessing the original content. Smart contracts are designed to be robust against collusion attacks, impersonation attacks, and tampering attacks.	SCM	Ethereum	Further research is required to utilize a wider variety of blockchain structural properties to maximize efficiency within SCM.	Gas cost, Computation cost
[31]	Proposed online pharma SCM. The model efficiently detects breaches during transportation using sensors, and payment is refunded to customers.	PSCM	Ethereum	The role of drug manufacturers, distributors, and consumers may be considered.	Execution cost, Transaction cost
[32]	Proof of delivery for physical assets has been adopted to reduce disputes in automatic payments involving single or multiple transporters. The model effectively prevents replay, denial-of-service, and man-in-the-middle attacks.	Transportation	Ethereum	Prototype design is required to evaluate secure and automated digital asset sale transactions.	Execution cost, Transaction cost
[33]	Proposed resource sharing for supply chain management based on smart contracts ensures reliability and data authenticity in supply networks and efficient resource utilization in networks with large outsourcing and production surpluses.	SCM	Ethereum	Visibility and accessibility for specific actors should be determined and separated.	Computational cost
[35]	An ESP32S2 device is used as a proof of concept to assess and confirm the proposed data storage protocol's functionality. A smart contract and a decentralized web application are intended to display and demonstrate sensor data gathered from the public blockchain.	SCM	PoC, Mythix	Challenges raised by stakeholders must be addressed. Enhanced security analysis is required.	Computation cost, Power consumption, Memory usage
[36]	Proposed a decentralized authentication framework for smart city. The ECC algorithm generates the key, and security goals are ensured through trust and authorization smart contracts.	SCM	Contiki, Multichain, COOJA	No discussion about cost analysis and vulnerability analysis.	Throughput, Access time Delay, Execution time
[37]	Proposed a supply inventory system based on blockchain that uses trusted environments and smart contracts. Large supply chains with several complicated transactions benefit from increased transparency and vendor coordination.	SCM	Ethereum	No discussion on security analysis.	Gas cost
[59]	Proposed secure SCM by implementing various cryptographic techniques. These techniques ensure secure, computationally efficient, fast verification and resistance to quantum attacks. Multiple phases are involved by using different smart contract functions.	SCM	Ethereum	No discussion about cost and vulnerability analysis.	Throughput, Processing time, Latency
[62]	Proposed blockchain-based counterfeit drug scheme utilizes IPFS for off-chain storage and PoW for consensus verification and block validation.	PSCM	Ethereum	No discussion about smart contract vulnerabilities.	Communication cost, Throughput, Computation cost

threats and classifies the data. Attribute-Based Encryption (ABC) schema allows access control based on attributes. Typically, it involves using a public key and a private key, where the private key is associated with specific characteristics. The private key decrypts data encrypted using the corresponding public key and attributes. CP-ABE demonstrated security against a chosen ciphertext attack. However, this model analyzes the system performance in terms of privacy, identification, and authentication.

Kaneriya and Patel [24] proposed a credential verification system for specific attributes without revealing the contents of other attributes. All the attributes are placed randomly in a tree, which leads to a complex situation where attackers can identify a particular entity. This schema addresses issues of record forgery, tampering, and time-consuming verification. However, it enhances latency, space complexity, and bandwidth. Bocek et al. [25] proposed a blockchain-based framework to secure IoT data by tracking the temperature settings of IoT-enabled containers. This process reduces expenses due to process optimization, automation, energy efficiency, and data-driven approaches, and data integrity and regulatory compliance are guaranteed. Blockchain attracts companies from various sectors with its ability to save costs, decentralize infrastructure, and minimize inefficiency. However, this requires handling the security of IoT sensor data within an access control system.

Rehan et al. [26] proposed an innovative HLF method with IoT sensors. These sensors provide real-time tracking, increased transparency, and secure temperature monitoring for perishable items. Distributed database through HLF promises quick transaction speeds and guarantees data integrity. Experiments indicate fast data transport (data compression, efficient routing algorithms, and optimized network protocols), block formation (grouping transactions into blocks, consensus algorithms, and block size management strategies), and network connectivity (peer-to-peer communication), with transaction times as low as 48 s. However, this model is a viable solution for safe, transparent, and efficient supply chain management, especially for temperature-sensitive items. Due to intricately supply networks, consumers frequently need information about pharmaceutical items. Aslam et al. [27] proposed a framework to improve customer trust and access information on the Ethereum blockchain. This schema improves efficiency and traceability by using smart contracts to track medications along the supply chain. However, more investigation is required for broader implementation of the real-time pharmaceutical industry. Table 2 highlights the blockchain-based approach's contributions to improving security and transparency in the pharmaceutical supply chain.

Globalization amplifies problems in supply chains, such as supplier relationships, pricing, product authenticity, transparency, and

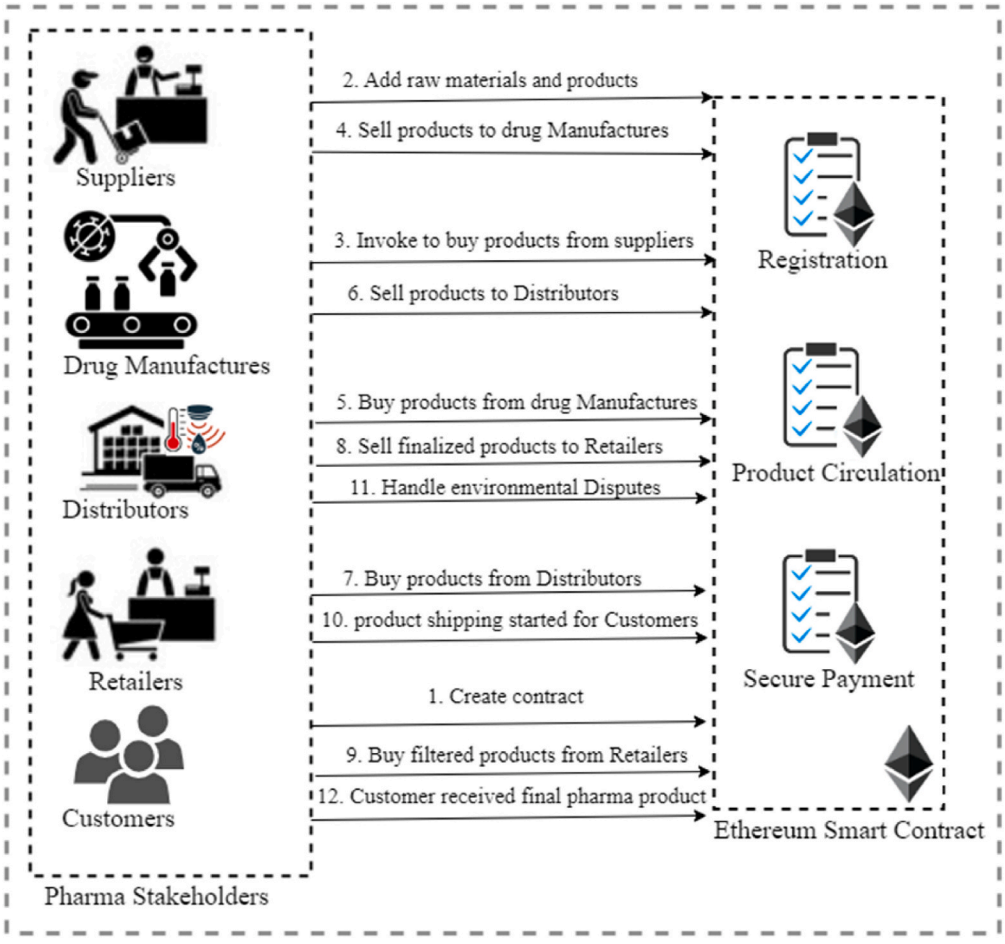


Fig. 1. Real world entities and workflow of the presented BPSCM system.

customer service. All these factors need careful consideration. The literature reviewed in this section highlights the importance of privacy aspects and stakeholder inclusion for ensuring reliable supply chain services. While existing frameworks analyzed here utilize cryptographic techniques and smart contracts to achieve authentication and transparency, a compelling need remains for a simpler, more cyber-resilient automated supply chain framework.

3. System model

This section presents the decentralized PSCM solution built upon the Ethereum platform blockchain to ensure authentication, product traceability, and product provenance. Fig. 1 describes the proposed model BPSCM solution along with its diverse stakeholders. To mitigate the technical difficulties incurred in logistics of the presented BPSCM includes suppliers (P_s), retailers (P_r), drug manufactures (P_{dm}), distributors (P_d), IoT containers (P_{IoT}), and consumers (P_c), such that entity identification is established through Ethereum account address. The description of the various building blocks of candidate BPSCM is summarized as follows.

- ➢ Suppliers (P_s): Companies act as suppliers, collecting raw materials and supplying them to drug manufactures.
- ➢ Drug manufactures (P_{dm}): Drug manufactures are companies that produce pharmaceutical drugs. They are responsible for formulating novel pharmaceutical compounds, engaging in large-scale production, and conducting comprehensive assessments to determine their safety and effectiveness.

Table 3
Smart contract functions intend to various stakeholders along with their privileges.

Function	Input	Output	Stakeholder involved	Description
Constructor()	NA	True/ False	P_r, P_{dm}, P_s	Initialize the stakeholders' Ethereum account addresses, parameters, states, and IPFS hash.
product_Add()	product_Name, product_Id, product_Quantity and product_Price	True/ False	P_r, P_{dm}, P_s, P_c	Add products/items on the BPSCM.
product_Remove()	product_Id	True/ False	P_r, P_{dm}, P_s, P_c	Remove the item from BPSCM
getnumberof_Products()	NA	Integer number	P_r, P_{dm}, P_s, P_c, P_d, P_{IoT}	Return the number of items ordered on request.
get_Product()	product_Id	Entity/ object	P_r, P_d, P_{IoT}	Returns information regarding the item for authentication.
order_Placed()	Order_Id, product_Id, product_Quantity, customer address	True/ False	P_r, P_{dm}, P_s, P_c, P_d	Orders are placed based on consumer requests, and other stakeholders are also indirectly involved.
is_Orderfullfilled()	product_Id	True/ False	P_r, P_{dm}, P_s, P_c, P_d	Returns sufficient items are their to full fill the order. If an order has already been placed, it returns false; otherwise, it returns true by checking the availability of items.
wholesaler_Add() and wholesaler_Remove()	Wholesaler address	True/ False	P_d	Based on the request, the number of wholesalers will be added and removed on BPSCM.
retailer_Add() and retailer_Remove()	Retailer address	True/ False	P_r, P_d	returns number of retailers will added and removed on BPSCM based on the request.
pharma_MAdd() and pharma_MRemove()	Pharma manufacture address	True/ False	P_{dm}, P_d	Add the pharma manufacture and remove based on request on BPSCM
quality_Control()	Address of quality control	True/ False	P_{dm}	Check the quality of the product and return true/false accordingly.
env_Datarecord()	Container_Id, temperature, humidity, timestamp	Entity/ object	P_{IoT}	Record the environmental change data in IoT container.
location_Record()	Container_Id, latitude, longitude, timestamp	Entity/ object	P_{IoT}	Record the location details of an authorized entity.
payment_Order()	order_Id, product_Id, product_Quantity, product_Price, supplier address, and customer address	True/ False	P_r, P_{dm}, P_s, P_c, P_d	A payment order was generated for a customer for product delivery.
deliver_Order()	order_Id	True/ False	P_r, P_c	Return order is delivered or not.
order_Dispute()	order_Id	String	P_r, P_{dm}, P_s, P_c, P_d	Returns the string value to indicate the order dispute.
refund_Processed()	order_Id, amount	Entity/ object	P_r, P_{dm}, P_s, P_c, P_d	The refund status was generated and received by a customer.
change_Ownership()	Ethereum address of new owner and old owner	True/ False	P_r, P_{dm}, P_s, P_d	Current owner has changed, and this function is accessible to the current stakeholder owner.
update_Stakeholder()	address	Integer number	P_r, P_{dm}, P_s, P_c, P_d, P_{IoT}	Update the stakeholder address to avoid the already existing account address.
upload_IPFShash()	item_Id, IPFS hash	hash	NA	Item hash value uploaded for auditing purposes.
get_Trackrecord()	trace_Id	Entity/ object	P_r, P_c, P_{IoT}	Get the tracking information about the product delivery.

- ➢ Distributors (P_d): Distributors sell the products to retailers. They often store the products in warehouses before shipping them to retailers.
- ➢ Retailers (P_r): After receiving the product from a drug manufacturer or distributor, retailers set reasonable pricing policies and sell the product to consumers.
- ➢ IoT container (P_{IoT}): Pharma products are delivered through containers equipped with IoT sensors to track and trace the product provenance record (delivery). IoT containers are equipped with sensors (temperature sensor, pressure sensor, GPS tracker) to track the temperature of pharma products, pressure due to accident jerk or opening of the container, and location of the container [25, 31].
- ➢ Customer (P_c): Customers purchase the products provided by the retailers; then, legitimate customers pay for products at a specific price if there are no product violations or damage. In BPSCM, customers can check the validity of products in the blockchain.
- ➢ IPFS server: All product details hash codes are uploaded to the IPFS server to reduce the blockchain's storage burden.

3.1. Threat model

The objective of an adversary seeking unauthorized access is to disrupt or monitor the distribution of pharmaceutical items. Adversaries are part of customers in the supply chain, and they may collude with the IPFS server to access product details stored in the server. Once the adversary gains access to this information, it could be altered and used for other purposes. Adversaries attempt to obtain access to the SCM by posing as legitimate users. Once they have access, they try to learn as much about the system's operations as possible. It would capture the product provenance record and supply chain [9,13].

3.2. Security goals

Our meticulously designed BPSCM model withstands attacks and fulfils crucial security and privacy objectives.

- ➢ Integrity and confidentiality: In the BPSCM model, all stakeholders uphold a product provenance record through a hash of their identity and password. The use of EdDSA to generate public and private keys ensures authentication and integrity. Product provenance is implemented to be resilient against various attacks such as forgery, tampering, and impersonation.
- ➢ Secured payment: Ensuring secure payment for stakeholders is a pivotal and often challenging aspect of the SCM system. In the BPSCM model, stakeholders can pay for products after verifying a proper authentication key generated using keccak256. Deploying a smart contract on the blockchain adds more security to this process.
- ➢ Record Auditability: In BPSCM, IoT sensors are installed within the container to track products. The other stakeholders know the product's shipping status without accessing the product's contents. This ensures that a secure hash algorithm and a Merkle tree storage structure for blockchain are used.

Our proposed system ensures privacy aspects of the product's provenance record, allowing anyone to verify its authenticity without access to its contents. It also enables stakeholders to track the product's journey through the supply chain tracking system and guarantees that all parties get compensated. Through this impressive set of security goals, our suggested approach can significantly enhance supply chain security. Table 3 outlines the deployed smart contract functions and their key characteristics.

4. Proposed BPSCM

This section presents the details of the proposed BPSCM, including various phases, such as registration of all stakeholders (P_s,P_r, P_{dm} P_d P_{IoT}, P_c), circulation of pharma products (product services, quality control and maintenance), and secured payment between stakeholders. The proposed framework maintains the pharma product information among the diverse stakeholders to ensure the appropriateness in delivering the final product to the end user. Here, we mainly focus on maintaining the confidentiality of the product's provenance record and preventing its information from being modified illegally throughout its circulation using symmetric encryption and EdDSA. EdDSA is faster, more secure against attacks, and requires less processing time than other cryptographic algorithms. The flow of the presented model is as follows.

- ➢ Initially, every stakeholder undergoes a registration process, and their identities are broadcast to other entities (Algorithm1).
- ➢ Pharma product details are encrypted using symmetric encryption, and the product provenance record is verified through the EdDSA technique.
- ➢ Encrypted pharma product details are shared among stakeholders (Algorithms 2 to 5), and environmental deviations are alerted to the appropriate entity.
- ➢ After product delivery, payment is established among stakeholders (Algorithm 6)
- ➢ All the transactions are validated and added to the blockchain using PoS Consensus.

EdDSA is a variant of the Schnorr signature method based on Elliptic Curve Cryptography (ECC). Ed25519 and Ed448 are variations of EdDSA based on Edwards curves. The experiment considered Ed25519, which generates keys using Curve25519 and SHA-512. This algorithm develops a key pair of 256-bit keys and 512-bit hashes. These hash functions exhibit four key characteristics: one-way

functions (easy to compute a message's hash but infeasible to reverse), preimage resistance (finding another message with the same hash is computationally hard), second-preimage resistance (finding another message with the same hash for a given message is computationally impossible), and strong collision resistance (finding message pairs with the same hash is infeasible). This approach is faster and more secure against several cryptographic attacks than other public key cryptographic algorithms [57,58,60].

The private key (PRk_{si}) is a randomly generated hashed number during the encryption process, while the public key (PUk_{si}) is derived from the private key. There are two forms of EdDSA, according to NIST and IRTF standards [60], depending on how the randomness is generated: $r_i = Prand(dk_i, M)$ in the first case and $r_i = Prand(dk_i, H(M))$ in the second. Where dk_i is derived key, M is a message, $Prand$ is a sudo random number, and H(M) is the hash of the message. In Ed25519, the PRk_{si} is randomly generated integer is further PUk_{si} is generated using a curve generator G_i on an elliptic curve as shown in equation (1) and the hash calculated by following equations [56].

$$PUk_{si} \leftarrow (PRk_{si} * G_i) \tag{1}$$

$$\mathrm{h} \leftarrow \{\mathrm{H}(r_i + PUk_{si} + M) mod Q\} \tag{2}$$

Where Q is a prime integer number

Then, the signature and r_i is calculated by using the following equations (3) and (4)

$$Sig_i \leftarrow (\mathrm{int} + h * PRk_{si}) \tag{3}$$

$$\{r_i, Sig_i\} \leftarrow E_{sign}\ (\mathrm{M}, PRk_{si}) \tag{4}$$

After generating the signature, the public key is used for verification. First, calculate the hash using equation (2), then calculate the verification points $\{P_1, P_2\}$ using equations (5) and (6), and finally verify that both verification points are equal $(P_1 = P_2)$. The verification of the signature is represented by equation (7).

$$P_1 \leftarrow (Sig_i, G_i) \tag{5}$$

$$P_2 \leftarrow (r_i + \mathrm{h} * PUk_{si}) \tag{6}$$

$$\frac{\mathrm{Valid}}{\mathrm{Invalid}} \leftarrow E_{verify}(\mathrm{M}, PUk_{si}, r_i, Sig_i) \tag{7}$$

In BPSCM the involved stakeholders are P_s, P_r, P_{dm} P_d P_{IoT}, P_c, and IPFS server. These stakeholders are represented as S_i, where $i \epsilon 1, 2, 3, 4, 5, 6$ and every stakeholder maintains the unique Ethereum account address (EA_{si}) along with PRk_{si} and PUk_{si} which are generated using EdDSA. The encrypted pharmaceutical details are initially stored in an IPFS server to preserve privacy and confidentiality.

4.1. Registration smart contract

In BPSCM, every stakeholder S_i maintains separate unique EA_{si} along with PRk_{si} and PUk_{si} to achieve secure authentication using smart contracts. Additionally, stakeholder S_i has to submit own password (Pwd_{si}), which is rendered using a random verification number and a hash of the private key when submitting the registration contract. And generated identity Id_{si} in equation (8) has broadcast to all stakeholders to resist the password guessing attack and the private information of stakeholders S_i will not be disclosed anywhere.

$$Id_{si} = S\_Register(Pwd_{si}, h(PRk_{si})) \tag{8}$$

Algorithm 1: Stakeholders Registration

Input: Pwd_{si}, PRk_{si}
Output: Identity Id_{si} of stakeholder
Procedure: Create a registration smart contract
    Generate Ethereum address EA_{si} for all stakeholders: $EA_{si} \leftarrow S_i$
    $S_i : S_i \leftarrow Pwd_{si}$, where $i \epsilon 1, 2, 3, 4, 5$
    $Id_{si} = S\_Register(Pwd_{si}, h(PRk_{si}))$
    Broadcast Id_i to all S_i
    State: state ← created
    Emit an event successfully generates identity Id_i for all stakeholders S_i
EndProcedure

4.2. Circulation of pharma products smart contract

In BPSCM, product information is shared with different stakeholders. We used the EdDSA signature schema and symmetric encryption to verify the product provenance record during pharmaceutical product circulation. This ensures the traceability of the product and protects it from unauthorized modifications by malicious entities. Fig. 2 illustrates the sequence flow of product

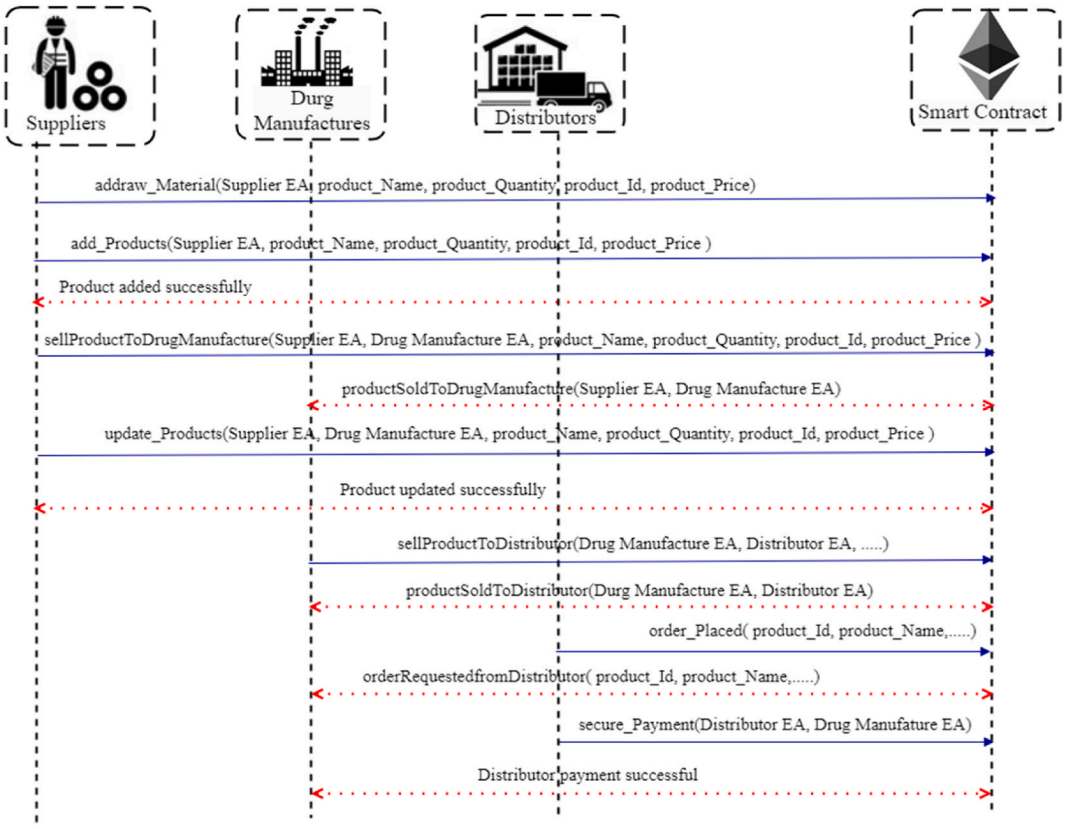


Fig. 2. Sequence diagram showing the interaction between supplier, drug manufactures, distributor and smart contract.

circulation between suppliers, drug manufacturers, and distributors.

Case1. Supplier to Drug manufactures

Suppliers provide the raw materials API to drug manufactures, they took the responsibility of generating the product provenance record R1, and it represented as equation 9

$$R_1 \leftarrow Id_{s1} \oplus Id_m \oplus Ai_{m1} \tag{9}$$

Where Id_{s1} represents the identity of suppliers, Id_{m1} specifies the raw material identity and Ai_m denotes the additional information about the pharma raw material. Suppliers encrypt the R_1 using an encryption key (k_1) which is generated by symmetric encryption (AES), the generated cipher text (T_{c1}) is $T_{c1} \leftarrow E(k_1, R_1)$. Now, suppliers' materials are protected from malicious activities by generating a hash for the product provenance record R1 as shown in equation 10

$$H(T_{c1}) \leftarrow PRk_{si} \; mod \; R_1 \tag{10}$$

$E_{sign1} \leftarrow H(T_{c1})$, where E_{sign1} denotes the signature of T_{c1} and $H(T_{c1})$ is the hash of cipher text. To access the record R_1 drug manufactures require the suppliers' public key (PUk_{si}) where i = 1. Algorithm 2 represents how smart contracts add products to suppliers, increasing stakeholder's visibility of existing stock products/items. A registered supplier smart contract is deployed once on the Ethereum blockchain and is responsible for maintaining track of the inventory system. If Suppliers want to add any product, they can specify the product_Name, product_Id, product_Quantity, and product_Price. Still, if the product already exists, the supplier makes the modifications on product_Quantity by entering the product_Id and updating the quantity sent to stakeholders to generate the event.

Algorithm 2: Adding and Removing products to Suppliers

Input: product_Name, product_Id, product_Quantity and product_Price
Output: Products added to suppliers (P_s)
Procedure: Modifier ← onlyOwner (suppliers (P_s))
suppliers (P_s) add_product details in supplier smart contract
Restrict access to $P_s \; e \; S_i$
If supplier = registered and product_Id already exists then

(*continued on next page*)

(*continued*)

Algorithm 2: Adding and Removing products to Suppliers

```
    Increase product_Quantity
  Else
      add_product by setting product_Name, product_Quantity and product_Price
  Endif
  If select product_Id to remove
      then product_Removed ← product_Id
  Else
    No such product_Id existed
  Endif
  Revert smart contract state to false
  Return number of products
  Broadcast the updates by triggering the event
EndProcedure
```

Case2. Drug manufactures to distributor

Drug manufactures buy products from suppliers and sell finalized products to distributors. Algorithm 3 presents the process of selling products from drug manufactures to distributors. Initially, the smart contract verifies that both entities are registered, agrees to the next product sale, and pays the purchase price. If both conditions are satisfied, then the contract state changes to productRequestAgreed. If the conditions are unsatisfied, the contract state is changed to productRequestFailed.

Algorithm 3: Distributor buy products from Drug manufactures

```
Input: Distributors Ethereum account address (EA_di), Drug Manufactures Ethereum account address (EA_dmi), product_Quantity and product_Price
Output: True/Flase
Procedure: Modifier ← onlyOwner (P_d)
  Modifier ← onlyOwner (P_dm)
  Contract state: buyfromDrugManufactures
  Restrict access to P_d ϵ S_i
  If distributor agreed terms and product_Price = paid then
    Contract state changed to productRequestAgreed
    Create notification message and send to P_d
  Else
    Contract state changed to productRequesrFailed
    Create notification message request failure
  Endif
  Revert smart contract state to false
  Return number of products
  Broadcast the updates by triggering the event
EndProcedure
```

Case3. Distributors and Drug Manufacture to Retailer

The retailer received the pharma products from either the distributor or drug manufactures and verified the product provenance record R_2 validity computed and R_2 represented as equation 11

$$R_2 \leftarrow Id_{r1} \oplus Id_{d1} \oplus Id_{dm1} \oplus Ai_{m2} \tag{11}$$

Where Id_{r1} represents the identity of a retailer, Id_{d1} specifies the identity of the distributor, Id_{dm1} represents the identity of the drug manufacture and Ai_{m2} represents the additional information about the pharma raw material. Distributors or drug manufactures encrypt the R_2 using an encryption key (k_2) which is generated by symmetric encryption, the generated cipher text (T_{c2}) is $T_{c2} \leftarrow E(k_2, R_2)$. Now, retailers are protected from malicious activities by generating a hash for the product provenance record R_2 as as shown in equation 12

$$H(T_{c2}) \leftarrow Sk_{si} \; mod \; R_2 \tag{12}$$

$E_{sign2} \leftarrow H(T_{c2})$, where E_{sign2} denotes the signature of T_{c2}. In algorithm 4, ordered products are shipped from distributor/drug manufactures to retailers. Initially, the smart contract verifies whether the product sale is agreed upon and payment has been successful, and then the smart contract states changes to saleRequestAgreed. If neither condition is met, the smart contract state is changed to saleRequestDenied. Fig. 3 illustrates the sequence flow of the drug manufactures, distributors and retailers with smart contracts.

Algorithm 4: Distributor and Drug Manufacture to Retailer

```
Input: Distributors Ethereum account address (EA_di), Drug manufactures Ethereum account address (EA_dmi), dateof_Manufacture, and product_Quantity,
Output: True/Flase
Procedure: Modifier ← onlyOwner (P_d)
```

(*continued on next page*)

(*continued*)

Algorithm 4: Distributor and Drug Manufacture to Retailer

```
Modifier ← onlyOwner (P_dm)
Modifier ← onlyOwner (P_r)
Contract state: productsoldtoDistributor/DrugManufacures
Retailer state: readytoPurchase
Restrict access to P_r ∈ S_i
If sale = agreed and payment (Algorithm 6) = successful then
   Contract state changed to saleRequestAgreed
   Emit success message
Else
   Contract state changed to saleRequestDenied
   Emit sale request failure message
Endif
Revert smart contract state to false
Broadcast the updates by triggering the event
EndProcedure
```

Case4. Distributor and Retailer to Consumer

Finally, the consumers buy the pharma products from retailers or distributors. Algorithm 5 presents the complete selling process from distributors/retailers. Initially, the smart contract checks whether payment is successful; if payment is successful, the contract state changes to productSoldtoCustomer; otherwise, it emits productSaleDenied. If the product is delivered on time, the contract states changes to productDeliveredOnTime; otherwise, return productDeliveryFailedOnTime. In the cases of dispute found due to container conditions and in-time delivery, authentication failed due to the appropriate hash key and successful delivery shown in Fig. 4. If the consumer fails to provide the proper secret code during product dispatch, a smart contract offers a 24-h time frame to provide the correct secret code; if the period exceeds, half the amount would be refunded to the consumer, and product delivery would be terminated. In each case, stakeholders S_i have to compute record R_i and generate cipher text T_{cti} to send the products to another stakeholder, they utilize the PRk_{si} to generate signatures and provenance records. At the end of each case, a transaction is created. To verify the validity of the provenance record by following the equation

$$E\left(E_{signi}, G_i\right) = E(H(T_{ct1} \oplus T_{ct2})) \tag{13}$$

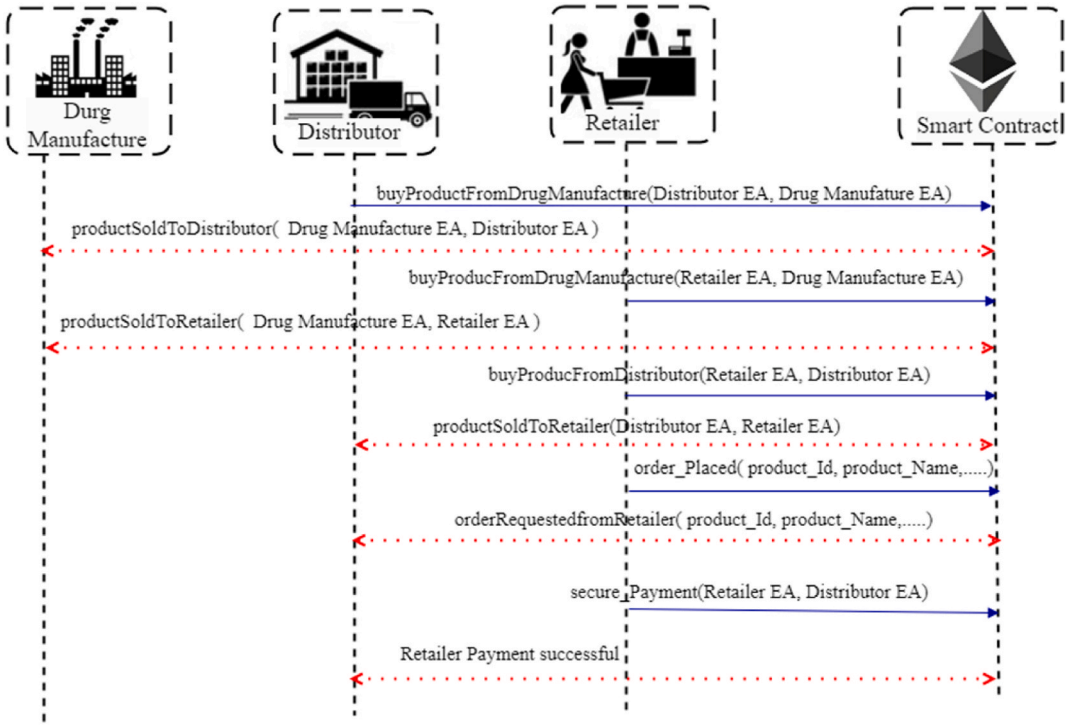


Fig. 3. Sequence diagram showing the interaction between drug manufactures, distributor, retailer and smart contract.

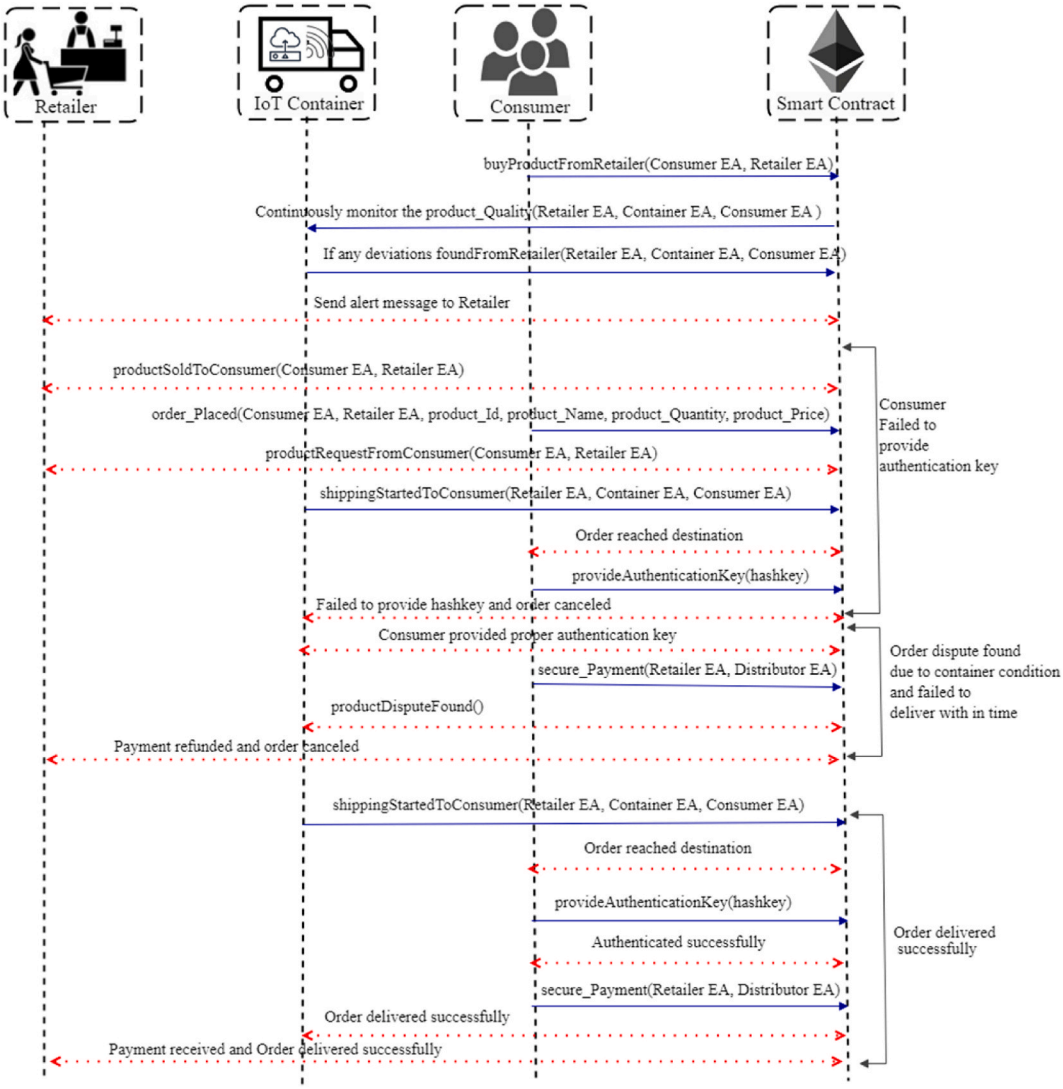


Fig. 4. Sequence diagram showing the interaction between retailer, consumer and smart contract.

Algorithm 5: Product Delivery Distributor and Retailer to Consumers

Input: Retailer Ethereum account address (EA_{ri}), Distributor Ethereum account address (EA_{di}), Customer Ethereum account address (EA_{ci}), purchased_Date, product_Id, hashkey, current_Time, and delivery_Time
Output: Product delivered successfully
Procedure: Modifier ← onlyOwner (P_d)
  Modifier ← onlyOwner (P_r)
  Modifier ← onlyOwner (P_c)
  Contract state: productReadytoBuy
  Restrict access to $P_c \in S_i$
  If product_Payment (Algorithm 6) = successful then//to check payment
    Contract state changed to productSoldtoCustomer
    Emit purchase success message
  Else
    Contract state changed to productSaleDenied
    Emit purchase failure message
  Endif
  If current_Time < delivery_Time then//exceed delivery time
    Contract state changed to productDeliveredOnTime
    Emit order cancellation message

(*continued on next page*)

(*continued*)

Algorithm 5: Product Delivery Distributor and Retailer to Consumers

```
    Else
      Refund payment to stakeholder (S_i)
      Contract state changed to productDelivereryFailedOnTime
      Revert smart contract state to false
    Endif
    Broadcast the updates by triggering the event
EndProcedure
```

4.3. Secure payment smart contract

In BPSCM, the stakeholders settle the payment for raw materials and products delivered to various stakeholders. Stakeholders have established secure payment through smart contracts. In algorithm 6, customers are verified through the kecchak256 hash function, and upon the hashed key match with the stored hash, the contract state is changed to customerVerificationSuccess. The payment has begun, and the balance is sufficient to buy the product. Then, transfer the total calculated amount to stakeholders and return paymentSuccess contract state. If the stakeholder balance is insufficient, the payment fails and returns an insufficientBalance contract state. In BPSCM, the stakeholders S_i have invoked a secured payment smart contract. The consumer pays the stakeholder S_{i+1} for successful product delivery, and the expense is expressed as S_E. If the product delivery fails due to manufacturing or other issues, stakeholder S_i will refund stakeholder S_{i+1}, representing the refund as S_R.

Algorithm 6: Secure Payment among Stakeholders

```
Input: product_Price, product_Id, stakeholders Ethereum account address EA_si , where i ϵ 1,2,3,4
Output: Payment successfully
Procedure: Modifier ← onlyOwner (S_i)
  ontract state: ReadytoPayment
  Restrict access to P_c ϵ S_i
  Compute keccak256 hash key//customer verification process
  Z ← keccak256(key)
  If z = keyhash and check record provenance success using equation (13)
    Contract state changed to customerVerificationSuccess
    Start the payment process
    Total = Calculate the no.of.products * product_Price + transport_Fee
    If check balance is to pay total
      Transfer total payment to stakeholder (S_i)
      Contract state changed to paymentSuccess
      Emit product payment successful
  Else
    Balance is not sufficient to pay
    Contract state changed to insufficientBalance
  Endif
    Else
    Contract state changed to paymentFailure
    Emit payment failure message
  Endif
  Broadcast the updates by triggering the event
EndProcedure
```

Our model is robust enough to track, monitor, and verify product consistency during various situations. Smart contracts track the IoT sensors' data deviation and trigger alert messages, especially for temperature and pressure. Smart contracts ensure all transactions are tamper-proof within BPSCM. A secure encryption schema maintains the confidentiality of the product's provenance record and prevents its information from being modified illegally throughout its circulation. Use the hash value to verify the legitimacy of provenance records without disclosing their origins to investigators.

Ethereum 2.0 with PoS improves key performance indicators in the PSCM system, including scalability, security, and energy efficiency. PoS improves scalability by improving transaction throughput, which is essential for managing substantial data and interactions in PSCM. It also decreases latency by allowing faster block validation and transaction confirmation times. PoS strengthens security by making 51 % of attacks less practical, as holding the majority of staked tokens is expensive. Furthermore, PoS minimizes processing expenses and energy usage, resulting in lower gas fees and a more cost-effective system for all participants. Key aspects to consider are throughput, latency, gas consumption, and security against attacks [54].

5. Testing and validation

The candidate BPSCM model simulation was run on five systems equipped with an Intel i7 CPU, 16 GB of RAM, and 1 TB of SSD

capacity, running the Ubuntu 20.04 operating system and other necessary specifications to develop the entire architecture. These systems were designed explicitly as a peer-to-peer network of various entities (i.e., P_s, P_r, P_{dm}, P_d, P_c). Specifically, we use the local Ethereum 2.0, Ganache, Geth client, Remix, and Metamask to determine the effectiveness and sustainability of smart contracts intended for all operations and the supply chain [11]. The included PoS, known for its energy efficiency, ensures miners do not require complex mathematical computations to validate transactions. After deploying the smart contracts, byte code is generated, and blocks are validated through PoS consensus. This process ensures a fair distribution of block creation and validation of the blocks upon PoS [48], instilling confidence in the technology's reliability. The smart contracts, designed with user security in mind, use modifiers to ensure that only users with the stakeholder role can access them. The smart contract's state is meticulously checked before the algorithm's execution and updated to the next state upon completion. Users create events to alert modifications and verify the results by examining the transaction-specific logs stored in the application development environment. These logs contain detailed information about the transaction output, events, gas usage amount, and exceptions. The ability to identify errors in the IDE by debugging the deployed code further enhances the sense of security in the system.

Moreover, the development environment offers transaction gas restrictions that may also be changed by the developer, offering an accurate representation of the mainnet network in Ethereum. The pharma supply chain, a critical sector, grapples with significant social and economic concerns, physical security, and freight delivery and quality difficulties. The high cost of these challenges is a significant barrier to achieving rapid evolution [31], underscoring the urgency and importance of finding effective solutions. The sample code for authenticity is shown in Fig. 5.

In our testing scenario, all the smart contracts (Registration, Product Circulation, and Secure Payment) are deployed in the private Ethereum test network. Initially, all the stakeholders registered in the registration smart contract were assigned unique Ethereum addresses, and the 20-byte identifier was used to identify each stakeholder. The unique addresses of deployed contracts about diverse stakeholders are furnished in Table 4, along with gas consumption, transaction cost, and execution cost. Login() functionality extends seamlessly to all registered stakeholders without relying on their assigned roles. Upon the successful execution of the function without any errors, trigger an event notification to acknowledge smooth login completion. For instance, product_add() functionality is used by the supplier and drug_manufaturer to either include the product details or view them. Here, in Fig. 6, the snapshot shows the details of the successful inclusion of the product by the supplier.

If suppliers want to add any raw materials to facilitate the addition of new raw materials by suppliers to polish the pharma products, then the addraw_Material(1) function is invoked. This includes specific details such as product name and price. If this scenario is successful, the contract state transitions to rawMaterialAddedSuccessfully, and Fig. 7 visualizes the function execution logs to provide transparency into the raw material process.

The remainder of the accounts belong to suppliers who keep adding their products by using unique item numbers with available quantities. These events notify the vendors by setting reactions if any changes to the inventory. In the same way, another real-world entity, drug manufacturers, identify any disputes that happen in products, which leads to the OrderDisputeFound state. Fig. 8 shows the logs related to order disputes. The order dispute may occur due to the need to provide a proper authentication key, IoT container conditions, and delays in time. The product quality is checked in each phase to deliver the item; the product quality depends on the date of manufacturing medicines, the molecules involved, and the drug manufacturing company. If there is no compromise in product

```
// Function to provide a secret code
function provideSecretCode(string memory code) public {
    // Check freight state and breach type
    require(
        (freightstate == FreightState.WaitingforSecretCode || freightstate == FreightState.WaitingForCorrectCode) &&
            breach == BreachType.None,
        "Invalid state or breach detected"
    );
        acquiredCodeToBehashed = keccak256(abi.encodePacked(code));
    // Authentication check
    if (secrtcode == acquiredCodeToBehashed) {
        // Successful authentication
        freightstate = FreightState.AuthenticatedByConsumer;
        emit ConsumerSuccessfullyAuthenticated("SUCCESS: Secret code matched", msg.sender);
    } else {
        if (attemptsRemaining > 0) {
            freightstate = FreightState.WaitingForCorrectCode;
            attemptsRemaining--; // Decrement attempts
            emit ConsumerFailedAuthentication("Incorrect code. You have X attempts remaining", msg.sender);
        } else {
            // Handle exceeded attempts (e.g., lock account, report breach)
            freightstate = FreightState.BreachDetected; // Or a more specific breach state
            breach = BreachType.Minor; // Or a more appropriate breach type
            emit ConsumerFailedAuthentication("Too many failed attempts. Account locked", msg.sender);
        }
    }
```

Fig. 5. Smart contract code snippet for authenticity.

Table 4
Deployed smart contracts with a unique address.

Contract name	Ethereum account address	Gas	Transaction cost	Execution cost
Suppliers	0 × 5B38Da6a701c568545dCfcB03FcB875f56beddC4	1381041	1201239	1069053
Drug manufactures	0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2	1309535	1139054	1010690
Distributors	0 × 4B20993Bc481177ec7E8f571ceCaE8A9e22C02db	161209	140181	118697
Retailer	0x78731D3Ca6b7E34aC0F824c42a7cC18A495cabaB	1309494	1139018	1010698
Customer/Consumer	0x617F2E2fD72FD9D5503197092aC168c91465E7f2	1539385	1338969	1197987

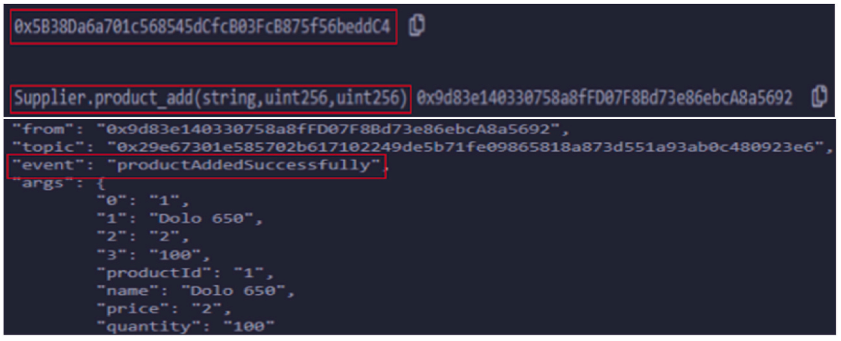


Fig. 6. Logs triggered upon submitting supplier to add products.

Fig. 7. Logs triggered upon submitting supplier to add raw material.

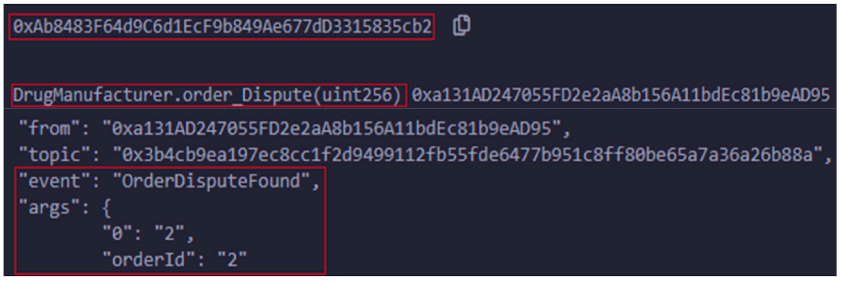


Fig. 8. Logs triggered upon submitting drug manufacture find order dispute.

quality, then the state transitions to QualityControlPassed. Fig. 9 shows the related logs for handling the distribution of products by distributors and retailers. After conducting a quality inspection, distributors successfully supply goods to retailers, distributing the final products to customers. Once all quality checks are completed, merchants deliver the products to consumers. Fig. 10 illustrates the successful delivery of orders by merchants.

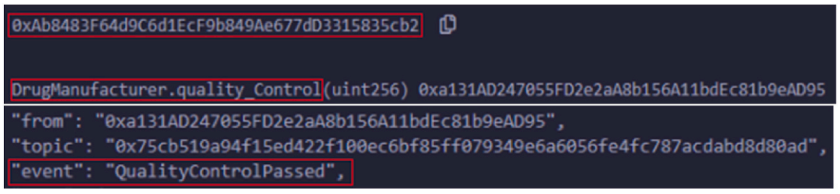


Fig. 9. Logs triggered upon submitting drug manufacture to check product quality.

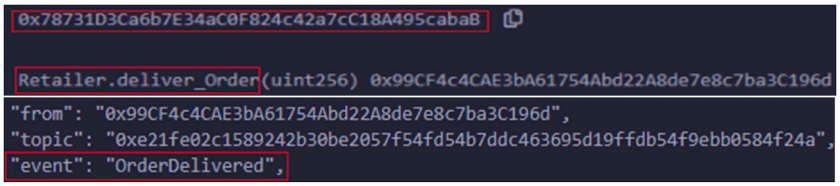


Fig. 10. Logs triggered upon submitting retailers order delivery.

5.1. Security analysis

This section examines potential vulnerabilities and security concerns in smart contracts within the proposed pharma supply chain solution. The presented model demonstrates its ability to ensure trust, data integrity, availability, non-repudiation, and cyber-attack resilience. Like any other complicated system, the Ethereum blockchain is vulnerable to various security concerns, including network and consensus attacks. Network attackers use DDoS and Sybil attacks to impede operations by exploiting communication flaws. Attackers using consensus techniques aim to compromise the fundamental validation procedure by attempting to manipulate transactions via majority control. Although research [43] highlights the promising solutions to Ethereum's security concerns, attention is still needed. Proactive security measures and constant assessment are required due to the growing scale of the network, the distribution of miners, and changing attack methods. Recognizing Ethereum's advantages and the constantly evolving possibility landscape enables developers, consumers, and other stakeholders to make well-informed decisions within this continually changing technical ecosystem. MyThril, an open-source program known for its outstanding performance in identifying security flaws in smart contracts, is utilized by the presented applications. MyThril is written in Python and executed using a command line tool [40]. The security analysis relies on contaminate analysis, dynamic symbolic execution, and control flow verification to examine the Ethereum Virtual Machine (EVM) bytecode for vulnerabilities. A Docker image of the tool was used to test smart contract vulnerability. The program carefully inspected files for flaws such as unchecked return values, assert violations, unprotected Ether withdrawals, delegate calls to untrusted callees, integer underflows, unauthorized storage writes, etc. [41,42]. The audit of the pharma contract produced a comforting result, as shown in Fig. 11. The analysis was completed successfully, and no issues were detected. These results reinforce our system's credibility by offering solid proof of the security of our suggested smart contract-based pharma supply chain.

However, timestamp dependencies (smart contracts that rely on timestamps are vulnerable to tampering, jeopardizing their functioning or security), transaction ordering dependencies (they arise when the result of one transaction depends on the sequence in which other transactions occur), integer overflows and underflows leads to contract's behavior may become unpredictable and susceptible to manipulation. The re-entrancy attack happens when a malicious contract actively calls back into the target contract before the first transaction is complete. This hole enables the attacker to repeatedly call the vulnerable function, draining funds before the contract can update its internal state. The experiment used acknowledged best practices for smart contract development to mitigate this. Specifically, we structured our smart contracts to conduct state changes before making any external calls, removing the possibility of external contracts violating re-entrancy flaws. Furthermore, modifiers to guarantee that the execution process is locked during crucial operations, eliminating recursive call loops that might jeopardize the contract's state integrity. Moreover, we used the Oyente tool, which is well-known for its in-depth analytical capabilities, for a more extensive vulnerability evaluation.

Oyente carefully examined the code to identify various EVM security flaws [44]. This comprehensive analysis resulted in a report summarizing EVM coverage with 60.2 %, which means Oyente significantly analyzed byte code instructions more than 50 % and a critical "false" decision for all reported vulnerabilities, indicating high security in our code, as shown in Table 5. Oyente is compatible with several older versions of the Solidity compiler, which is crucial for understanding. As a result, before the study, code syntax

Fig. 11. Smart contract Vulnerability analysis report using MyThril tool.

Table 5
Smart contract Vulnerability analysis report using Oyente tool.

Parameter	Results
EVM Code Coverage	60.2 %
Integer Underflow	False
Integer Overflow	False
Timestamp Dependency	False
Re-entrancy Vulnerability	False
Transaction ordering dependency (TOD)	False
Parity multi-sig Bug	False

changes were required to comply with these restrictions. Moreover, Oyente offers insightful direction and line-by-line suggestions for fixing any possible flaws, enabling iterative improvement and optimization of our code. This meticulous process, which included intelligent error detection and extensive security research, gave us confidence in the resilience and security of our smart contract usage.

The BPSCM uses smart contract capabilities to manage authentication and access control, with restricted modifiers allowing registered stakeholders to execute operations. In addition, the system hashed every product detail with a timestamp. The EdDSA version we used in BPSCM is unforgivable by MITM (Man-in-the-Middle). Thus, the system prevents **Replay** and **MITM** attacks. Since only the legitimate user can access the private key, the attacker cannot access the product, even if he replaces it with his own EA and public key. Therefore, the system would reject the transaction and reverse the contract state if an attacker were to impersonate a legitimate user's IP address or EA address. **Integrity** ensures no one alters data to exclude key information by using logs and creating events. Our model can monitor and trace the history of product details. The Ethereum address of the function call initiator is always captured and included in the logs, and **non-repudiation** confirms the stakeholder's identity. It guarantees that the user cannot retract their actions. The smart contract functions are executed, and logs are available to stakeholders to ensure their availability. It provides all services are available to users and protects against **Denial of Service** (DoS) due to an immutable public ledger.

The publicly available private Ethereum blockchain was chosen for BPSCM because it offers anonymity to all stakeholders, even though private blockchains like Multichain, Hyperledger, and private Ethereum offer transaction encryption and restricted access control. It provides pseudonymity, and the blockchain's unchangeable record of timestamped transactions encourages supply chain accountability and transparency. BPSCB ensures the **confidentiality** by using symmetric encryption. This openness makes it possible for all parties to be informed about inventory levels and makes it easier to respond quickly to unforeseen changes in demand. IPFS hash stores product information to reduce the storage burden. The product provenance record is generated through unique stakeholder identification with XoR operations and broadcasted to the network. Initially, we encrypted the raw material using symmetric encryption and performed record verification through EdDSA. It is difficult for adversaries to tamper with product information. This ensures that our model BPSCM is resistant to **tampering attacks**. A **key guessing attack** is similar to a brute-force attack, which involves a cryptanalyst attempting all possible guesses for the private key. The suggested approach generates 256-bit long private keys using the SHA-256 hash algorithm. Hence, a brute force assault cannot compromise the security of the proposed system in a finite amount of time [56].

As previously stated, the product provenance record would be hashed into 256-bit codes to ensure authenticity during pharma product circulation. It guarantees the product's traceability and prevents unwanted entities from forbidden alterations. As a result, not all stakeholders can change or retrieve the hash value. Thus, our BPSCM successfully threw the **collusion attack**. If external adversaries succeed, password-guessing attacks can compromise stakeholder identities. We generate high-security identities that utilize the random verification number and hash of private keys and leverage smart contracts for secure registration. This approach enhances security against **impersonation attacks**. Table 6 compares the security analysis of the existing model regarding the perception of the above content.

The presented approach uses protocols to ensure only legitimate parties validate the transaction and block creation using robust authentication implemented in the registration phase. Hence, the loophole for the possibility of triggering 51 % of attacks was prevented to a greater extent. Additionally, Ethereum 2.0's PoS protocol provides substantial benefits for securing a BPSCM system. Unlike PoW, PoS is based on economic incentives, making it financially difficult for attackers to take control. Also, reducing penalties prevents malicious behaviour by penalizing validators who attempt to disrupt the network. In addition, PoS features like checkpointing and a safety oracle improve security by restricting the time frame for blockchain reorganization and identifying discrepancies [54].

Table 6
Comparision of security analysis with state-of art techniques.

Security Property	[30]	[31]	[33]	[56]	Proposed model
Collusion attack	Ć	×	×	Ć	Ć
Tampering attack	Ć	×	×	Ć	Ć
Key guessing attack	Ć	Ć	Ć	Ć	Ć
Impersonation attack	×	×	×	×	Ć
DDoS	×	×	×	×	Ć
Re-entrancy attack	×	×	×	×	Ć

These combined considerations make a 51 % attack on Ethereum 2.0 exceedingly difficult and economically unfeasible in the context of a PSCM. The decentralized structure of PoS and the financial disincentives for malicious behaviour offers a strong and secure foundation for securing sensitive pharmaceutical data and preserving supply integrity.

6. Performance evaluation

This section evaluates the proposed scheme's effectiveness and efficiency through various performance metrics. This assessment employs gas consumption, throughput, latency, and computational cost. When submitted to the private Ethereum blockchain, a minimum gas price is needed to cover a transaction's processing costs. Conveniently, Remix IDE's output interface estimates this cost for every transaction. Fig. 12 lists the expenses for the different smart contract code functions. As shown in the black line, the computing requirement of invoked operations indicates execution costs. Higher execution costs are a natural result of more sophisticated functions. Conversely, the red line represents transaction costs, including deploying the complete smart contract code into the blockchain. This is independent of code size and function, in contrast to execution cost. Manufacturers can optimize smart contracts cost-effectively and efficiently by identifying these different expenses [31,32].

6.1. Scalability

Scalability is crucial when implementing blockchain in PSCM, mainly due to the large volume of data generated. However, to enhance scalability, layer-2 solutions, namely, state channels or sidechains (to allow off-chain transactions and reduce the burden on the main blockchain) and sharding (splits a blockchain network into smaller partitions called shards to process more transactions), are implemented [61]. In our BPSCM, only approved participants can validate transactions, significantly improving the network performance. The proposed framework utilizes IPFS to store the hashed data instead of raw data in the immutable ledger. Due to hash values, the data fetches quickly, improving blockchain response time. Fig. 13 provides precise information regarding the blockchain based on IPFS, which is more scalable than the traditional baseline models AIBC [45].

6.2. Computation cost

To evaluate the computation cost of the validation process in a real-world environment, we mainly focus on encrypting the product details using a symmetric algorithm, hashing and EdDSA to verify user integrity and product verification [30]. In our framework, validation is carried out through suppliers, distributors, retailers, and drug manufacturers. Fig. 14 demonstrates the effectiveness of our approach, assuming a distinct number of supply chains (each with a supplier, distributor, retailer, and drug manufacturer) concerning computation cost. For instance, authenticating a single product record takes 1.4 s, takes 1.5 s, takes 1.6 s, and Pr takes 1.8 s for one single supply chain. Our BPSCM offers affordable computing costs and applicability for various use cases [28].

6.3. Throughput

Smart contracts can be deployed, executed, and invoked at different rates across blockchain platforms. Thus, it is vital to monitor transaction throughput, typically calculated as the rate at which the blockchain network conducts legitimate transactions during a certain period. The throughput shows the number of transactions completed per second, which is calculated using equation (14) [51].

$$\mathrm{T} = \frac{C(T_{tx}\ in\ (T_b, T_i))}{T_b - T_i} \tag{14}$$

Where, T is transaction throughput, C is count, T_{tx} is the total number of submitted transactions, T_b is block confirmation time, T_i transaction submission time. Fig. 15 shows that the transaction initiation time is constant regardless of the number of users. We vary the number of users from 1000 to 5000 until 4000; throughput variation is much less at 5000 users. Throughput drastically increases for the proposed and existing models. It also shows that our approach performs better than the existing models DSCMR [22] and

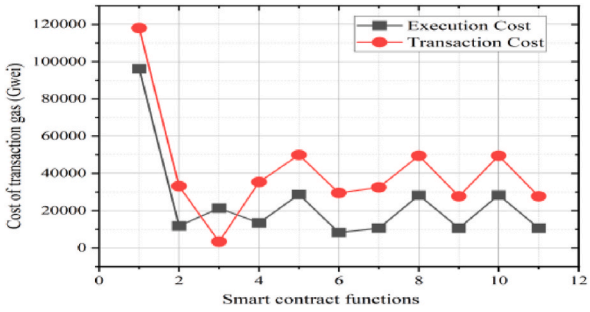


Fig. 12. Performance analysis of Pharma smart contracts.

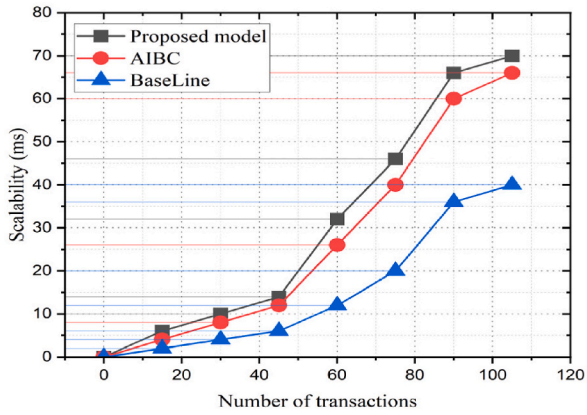


Fig. 13. Scalability analysis.

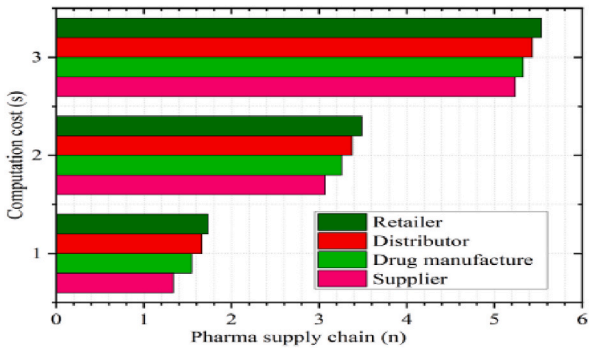


Fig. 14. Computation cost analysis.

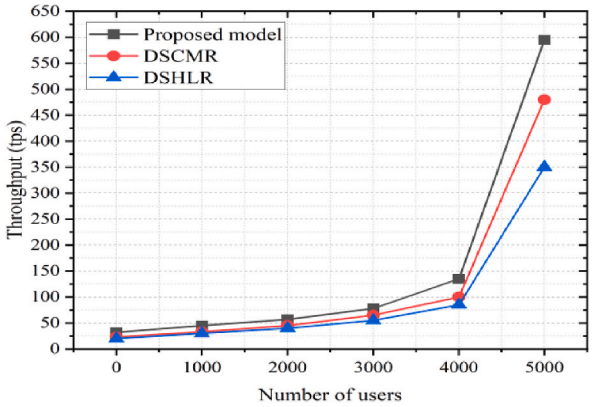


Fig. 15. Throughput analysis.

DSHLR [46].

6.4. Latency

The system requires a certain period to validate a transaction before it is sent to the network. Transaction latency is when a transaction is entered, confirmed, and committed, and the result is made available to every user on the network. In BPSCM, various factors impact latency, including block generation time, network propagation delay, and transaction validation time. Our solution is designed to minimize latency through efficient block formation and transaction processing processes. This statistic is measured for each transaction in terms of seconds and expressed as equation (15) [51].

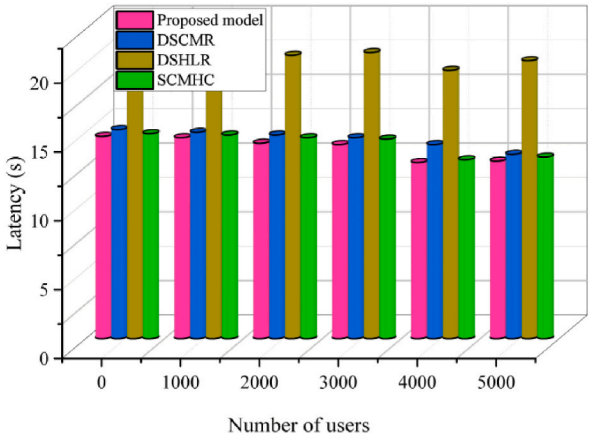


Fig. 16. Latency analysis to invoke the transactions.

$$L = \frac{\sum T_{tx}\left(T_{confirm} - T_{input}\right)}{\mathrm{C}(T_{tx}\ in\ ((T_b, T_i))} \tag{15}$$

Where, L is latency, $T_{confirm}$ is transaction confirmation and T_{input} is transaction input. The proposed BPSCM model for the pharmaceutical supply chain outperforms compared to BIVSC [47], PharmaChain [11] and ESCPSC [31] in terms of average processing time. Fig. 17 highlights this benefit in terms of performance, whereby by increasing the number of transactions for shipping products, the average transaction latency is low compared to existing models. For 100 average number of transactions, it records the low transaction delay compared to existing models [11,31,47]. After that, the latency grows linearly, although it still has a significant advantage over the examined models. However, this demonstrates the enhanced effectiveness of our proposed framework with varying numbers of transaction requests. On the other hand, to prove the effectiveness of our BPSCM approach, we considered latency analysis to invoke transaction efficiency by changing the number of users from 1000 to 5000. in this case, our strategy also yields enhanced improvement compared to existing models SCMHC [19], DSCMR [22], and DSHLR [46], as shown in Fig. 16 and 17.

6.5. Qualitative comparison

Table 7 summarizes the test results alongside those of other relevant solutions. This comparison aims to showcase the proposed solution's advantages within the pharmaceutical supply chain. The efforts undertaken in this work ensure the reliability and security of the proposed solution for the intended task. The suggested BPSCM decentralized approach eliminates the requirement for extra offline storage compared to the frameworks presented in Refs. [28–31,33–47]. Additionally, it offers DApps compatibility, a feature that must be added to existing solutions. This DApps compatibility fosters product sharing and reduces the risk of single points of failure. Additionally, the network operates independently, eliminating reliance on external entities like government agencies or developers for transaction validation.

7. Conclusion

This paper presented a novel blockchain-based PSCM framework that addresses the critical challenges of privacy, authentication, and data provenance in traditional online systems. The proposed solution leverages smart contracts and cryptographic mechanisms to ensure secure and transparent tracking of pharmaceutical products throughout the supply chain lifecycle. Real-world entities are integrated into the framework, enabling real-time monitoring and automated alerts for potential quality deviations. The proposed BPSCM is implemented in three phases: registration, product circulation, and secured smart payment contracts. For security enhancement, EdDSA and symmetric encryption have been included for product provenance record verification. Security analysis demonstrates that the framework can withstand various cyber-attacks, and smart contract vulnerability analysis has been established. Finally, in terms of efficacy, the tested results show improvements in the metrics of throughput, reduced latency, and computation cost compared to state-of-the-art models. In the future, we plan to integrate Artificial Intelligence techniques for drug recommendation and analysis. More focus on scalability issues and integration of cutting-edge technology.

CRediT authorship contribution statement

Adla Padma: Writing – original draft, Software, Resources, Methodology, Conceptualization. **Mangayarkarasi Ramaiah:** Writing – review & editing, Validation, Investigation, Formal analysis.

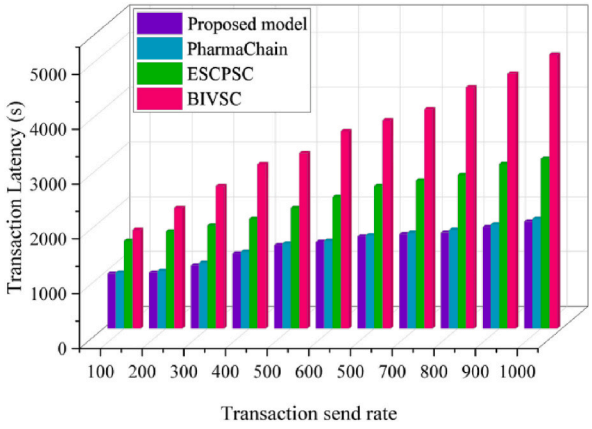


Fig. 17. Transaction latency analysis.

Table 7
Comparison to the cutting-edge technology.

Ref	Security Analysis using tool	Cryptography Technique	Storage	BC Platform	Cost analysis	Dapp
[28]	×	×	×	Ć	Ć	×
[29]	×	×	×	Ć	NA	Ć
[30]	Ć	Ć	×	Ć	Ć	×
[31]	×	×	×	Ć	Ć	×
[32]	×	×	Ć	Ć	Ć	Ć
[33]	×	×	×	Ć	×	Ć
[35]	Ć	Ć	×	Ć	×	×
[36]	Ć	Ć	×	Ć	NA	×
[37]	×	×	×	Ć	Ć	Ć
Our Work	Ć	Ć	Ć	Ć	Ć	Ć

Data availability

Data will be made available on request.

Declaration of competing interest

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

References

[1] D. Power, Supply chain management integration and implementation: a literature review, Supply Chain Manag.: an International journal 10 (4) (2005) 252–263.
[2] A. Wieland, Dancing the supply chain: toward transformative supply chain management, J. Supply Chain Manag. 57 (1) (2021) 58–73.
[3] Guardian, 10% of Drugs in Poor Countries Are Fake, Says WHO, 2017 [Online]. Available: https://www.theguardian.com/global-development/2017/nov/28/10-ofdrugs-in-poor-countries-are-fake-says-who. (Accessed 3 June 2020).
[4] L. Sparks, Supply chain management and retailing, Supply Chain Forum Int. J. 11 (4) (2010, January) 4–12. Taylor & Francis.
[5] S. Mabizela, H.N. Nakambale, V. Bangalee, Evaluation of Pharmaceutical Inventory Management Challenges at Public Healthcare Facilities in King Cetshwayo District, KwaZulu-Natal, 2023. South Africa.
[6] S.T. Mohammed, J.A. Hussien, A traceable and reliable electronic supply chain system based on blockchain technology, UHD Journal of Science and Technology 4 (2) (2020) 132–140.
[7] M.M. Ali, M.Z. Babai, J.E. Boylan, A.A. Syntetos, Supply chain forecasting when information is not shared, Eur. J. Oper. Res. 260 (3) (2017) 984–994.
[8] M. Lou, X. Dong, Z. Cao, J. Shen, SESCF: a secure and efficient supply chain framework via blockchain-based smart contracts, Secur. Commun. Network. 2021 (2021) 1–18.
[9] S.K. Dwivedi, R. Amin, S. Vollala, Blockchain based secured information sharing protocol in supply chain management system with key distribution mechanism, J. Inf. Secur. Appl. 54 (2020) 102554.
[10] M. Majdalawieh, N. Nizamuddin, M. Alaraj, S. Khan, A. Bani-Hani, Blockchain-based solution for secure and transparent food supply chain network, Peer-to-Peer Networking and Applications 14 (2021) 3831–3850.
[11] R. Mishra, D. Ramesh, N. Mohammad, B. Mondal, Blockchain enabled secure pharmaceutical supply chain framework with traceability: An efficient searchable pharmachain approach, Cluster Comput. 27 (2024) 13621–13641.
[12] R.K. Jha, P. Alam, N. Priyadarshi, M.A. Ghazi, M.S. Bhargavi, Counterfeit drug prevention in pharma supply chain using blockchain technology, in: 2023 3rd International Conference on Intelligent Communication and Computational Techniques (ICCT), 2023, January, pp. 1–6. IEEE.
[13] S. Al-Farsi, H. Bensmail, S. Bakiras, Securing blockchain-based supply chain workflow against internal and external attacks, Machines 10 (6) (2022) 431.
[14] C. Ma, S. Liu, G. Xu, HGAT: smart contract vulnerability detection method based on hierarchical graph attention network, J. Cloud Comput. 12 (1) (2023) 1–13.

[15] G. Subramanian, A.S. Thampy, N.V. Ugwuoke, B. Ramnani, Crypto pharmacy–digital medicine: a mobile application integrated with hybrid blockchain to tackle the issues in pharma supply chain, IEEE Open Journal of the Computer Society 2 (2021) 26–37.
[16] R.R. Konapure, S.D. Nawale, Smart contract system architecture for pharma supply chain, in: 2022 International Conference on IoT and Blockchain Technology (ICIBT), 2022, May, pp. 1–5. IEEE.
[17] A.K. Bapatla, S.P. Mohanty, E. Kougianos, D. Puthal, A. Bapatla, PharmaChain: a blockchain to ensure counterfeit-free pharmaceutical supply chain, IET Netw. 12 (2) (2023) 53–76.
[18] M. Uddin, Blockchain Medledger: hyperledger fabric enabled drug traceability system for counterfeit drugs in pharmaceutical industry, Int. J. Pharm. 597 (2021) 120235.
[19] D. Agrawal, S. Minocha, S. Namasudra, A.H. Gandomi, A robust drug recall supply chain management system using hyperledger blockchain ecosystem, Comput. Biol. Med. 140 (2022) 105100.
[20] A. Musamih, K. Salah, R. Jayaraman, J. Arshad, M. Debe, Y. Al-Hammadi, S. Ellahham, A blockchain-based approach for drug traceability in healthcare supply chain, IEEE Access 9 (2021) 9728–9743.
[21] F. Chiacchio, D. D'Urso, L.M. Oliveri, A. Spitaleri, C. Spampinato, D. Giordano, A non-fungible token solution for the track and trace of pharmaceutical supply chain, Appl. Sci. 12 (8) (2022) 4019.
[22] K. Abbas, M. Afaq, T. Ahmed Khan, W.C. Song, A blockchain and machine learning-based drug supply chain management and recommendation system for smart pharmaceutical industry, Electronics 9 (5) (2020) 852.
[23] R. Gaur, S. Prakash, S. Kumar, K. Abhishek, M. Msahli, A. Wahid, A machine-learning–blockchain-based authentication using smart contracts for an IoHT system, Sensors 22 (23) (2022) 9074.
[24] J. Kaneriya, H. Patel, A secure and privacy-preserving student credential verification system using blockchain technology, International Journal of Information and Education Technology 13 (8) (2023).
[25] T. Bocek, B.B. Rodrigues, T. Strasser, B. Stiller, 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, May, pp. 772–777.
[26] M. Rehan, A.R. Javed, N. Kryvinska, T.R. Gadekallu, G. Srivastava, Z. Jalil, Supply chain management using an industrial internet of things hyperledger fabric network, HUMAN-CENTRIC COMPUTING AND INFORMATION SCIENCES 13 (2023).
[27] M. Aslam, S. Jabbar, Q. Abbas, M. Albathan, A. Hussain, U. Raza, Leveraging Ethereum platform for development of efficient tractability system in pharmaceutical supply chain, Systems 11 (4) (2023) 202.
[28] Z. Raza, I.U. Haq, M. Muneeb, Agri-4-All: a framework for blockchain based agricultural food supply chains in the era of fourth industrial revolution, IEEE Access 11 (2023) 29851–29867.
[29] S. Balasubramanian, I.S. Akila, Blockchain implementation for agricultural food supply Chain using hyperledger fabric, J. Intell. Fuzzy Syst. 43 (5) (2022) 5387–5398.
[30] G. Zhang, Z. Yang, W. Liu, Blockchain-based decentralized supply chain system with secure information sharing, Comput. Ind. Eng. (2023) 109392.
[31] S. Abdallah, N. Nizamuddin, Blockchain-based solution for pharma supply chain industry, Comput. Ind. Eng. 177 (2023) 108997.
[32] H.R. Hasan, K. Salah, Blockchain-based proof of delivery of physical assets with single and multiple transporters, IEEE Access 6 (2018) 46781–46793.
[33] T.K. Agrawal, J. Angelis, W.A. Khilji, R. Kalaiarasan, M. Wiktorsson, Demonstration of a blockchain-based framework using smart contracts for supply chain collaboration, Int. J. Prod. Res. 61 (5) (2023) 1497–1516.
[34] K.C. Bandhu, R. Litoriya, P. Lowanshi, M. Jindal, L. Chouhan, S. Jain, Making drug supply chain secure traceable and efficient: a Blockchain and smart contract based implementation, Multimed. Tool. Appl. 82 (15) (2023) 23541–23568.
[35] J.H. Khor, M. Sidorov, M.T. Ong, S.Y. Chua, Public blockchain-based data integrity verification for low-power IoT devices, IEEE Internet Things J. 23 (2023) 13056–13064.
[36] S. Siddiqui, S. Hameed, S.A. Shah, A.K. Khan, A. Aneiba, Smart contract-based security architecture for collaborative services in municipal smart cities, J. Syst. Architect. 135 (2023) 102802.
[37] I.A. Omar, R. Jayaraman, M.S. Debe, H.R. Hasan, K. Salah, M. Omar, Supply chain inventory sharing using ethereum blockchain and smart contracts, IEEE Access 10 (2021) 2345–2356.
[38] K. Christidis, M. Devetsikiotis, Blockchains and smart contracts for the internet of things, IEEE Access 4 (2016) 2292–2303.
[39] R. Gaur, S. Prakash, S. Kumar, K. Abhishek, M. Msahli, A. Wahid, A machine-learning–blockchain-based authentication using smart contracts for an IoHT system, Sensors 22 (23) (2022) 9074.
[40] https://github.com/Consensys/mythril.
[41] M. Di Angelo, G. Salzer, A survey of tools for analyzing ethereum smart contracts, in: 2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON), IEEE, 2019, April, pp. 69–78.
[42] A. Padma, R. Mangayarkarasi, Detecting security breaches on smart contracts through techniques and tools a brief review: applications and challenges, in: International Conference on Information and Management Engineering, Springer Nature Singapore, Singapore, 2022, December, pp. 361–369.
[43] J.H. Khor, M. Sidorov, P.Y. Woon, Public blockchains for resource-constrained IoT devices—a state-of-the-art survey, IEEE Internet Things J. 8 (15) (2021) 11960–11982.
[44] S.S. Kushwaha, S. Joshi, D. Singh, M. Kaur, H.N. Lee, Systematic review of security vulnerabilities in ethereum blockchain smart contract, IEEE Access 10 (2022) 6605–6621.
[45] A. Alabdulatif, I. Khalil, M. Saidur Rahman, Security of blockchain and AI-empowered smart healthcare: application-based analysis, Appl. Sci. 12 (21) (2022) 11039.
[46] F. Jamil, L. Hang, K. Kim, D. Kim, A novel medical blockchain model for drug supply chain integrity management in a smart hospital, Electronics 8 (5) (2019) 505.
[47] B. Yong, J. Shen, X. Liu, F. Li, H. Chen, Q. Zhou, An intelligent blockchain-based system for safe vaccine supply and supervision, Int. J. Inf. Manag. 52 (2020) 102024.
[48] R. Asif, S.R. Hassan, Shaping the future of ethereum: exploring energy consumption in proof-of-work and proof-of-stake consensus, Frontiers in Blockchain 6 (2023).
[49] H. Hasan, E. AlHadhrami, A. AlDhaheri, K. Salah, R. Jayaraman, Smart contract-based approach for efficient shipment management, Comput. Ind. Eng. 136 (2019) 149–159.
[50] T.K. Mackey, G. Nayyar, A review of existing and emerging digital technologies to combat the global trade in fake medicines, Expet Opin. Drug Saf. 16 (5) (2017) 587–602.
[51] H. Honar Pajooh, M.A. Rashid, F. Alam, S. Demidenko, Experimental performance analysis of a scalable distributed hyperledger fabric for a large-scale IoT testbed, Sensors 22 (13) (2022) 4868.
[52] M. Ramaiah, V. Chithanuru, A. Padma, V. Ravi, A review of security vulnerabilities in industry 4.0 application and the possible solutions using blockchain, Cyber Security Applications for Industry 4 0 (2022) 63–95.
[53] A. Padma, M. Ramaiah, Blockchain based an efficient and secure privacy preserved framework for smart cities, IEEE Access (2024) 21985–22002.
[54] S. Sayeed, H. Marco-Gisbert, Assessing blockchain consensus and security mechanisms against the 51% attack, Appl. Sci. 9 (9) (2019) 1788, https://doi.org/10.3390/app9091788.
[55] A. Nawaz, L. Wang, M. Irfan, T. Westerlund, Hyperledger sawtooth based supplychain traceability system for counterfeit drugs, Comput. Ind. Eng. (2024) 110021.
[56] G. Shankar, L.H. Ai-Farhani, P. Anitha Christy Angelin, P. Singh, A. Alqahtani, A. Singh, I.A. Samori, Improved multisignature scheme for authenticity of digital document in digital forensics using edward-curve digital signature algorithm, Secur. Commun. Network. 2023 (2023).

[57] G. Zhao, H. He, B. Di, J. Chu, StuChain: an efficient blockchain-based student e-portfolio platform integrating hybrid access control approach, Multimed. Tool. Appl. 83 (1) (2024) 227–251.
[58] S. Josefsson, I. Liusvaara, Edwards-curve digital signature algorithm (EdDSA), 2017.
[59] S. Datta, S. Namasudra, Blockchain-based smart contract model for securing healthcare transactions by using consumer electronics and mobile edge computing, IEEE Trans. Consum. Electron. 70 (1) (2024) 4026–4036.
[60] Q. Feng, K. Yang, K. Zhang, X. Wang, Y. Yu, X. Xie, D. He, Stateless Deterministic Multi-Party EdDSA Signatures with Low Communication, Cryptology ePrint Archive, 2024.
[61] A. Padma, M. Ramaiah, GLSBIoT: GWO-based enhancement for lightweight scalable blockchain for IoT with trust based consensus, Future Generat. Comput. Syst. 159 (2024) 64–76.
[62] S. Datta, S. Namasudra, Blockchain-based Secure and Scalable Supply Chain Management System to Prevent Drug Counterfeiting, Cluster Computing, 2024, pp. 1–18.
[63] P. Sharma, S. Namasudra, P. Lorenz, Blockchain-based cloud storage system with enhanced optimization and integrity preservation, in: ICC 2023-IEEE International Conference on Communications, 2023, May, pp. 3744–3749. IEEE.
[64] A. Gupta, S. Namasudra, A novel technique for accelerating live migration in cloud computing, Autom. Software Eng. 29 (1) (2022) 34.
[65] S. Namasudra, P. Sharma, Achieving a decentralized and secure cab sharing system using blockchain technology, IEEE Trans. Intell. Transport. Syst. 24 (12) (2022) 15568–15577.
[66] A. Yazdinejad, R.M. Parizi, A. Dehghantanha, Q. Zhang, K.K.R. Choo, An energy-efficient SDN controller architecture for IoT networks with blockchain-based security, IEEE Transactions on Services Computing 13 (4) (2020) 625–638.
[67] A. Yazdinejad, A. Dehghantanha, R.M. Parizi, M. Hammoudeh, H. Karimipour, G. Srivastava, Block hunter: federated learning for cyber threat hunting in blockchain-based iiot networks, IEEE Trans. Ind. Inf. 18 (11) (2022) 8356–8366.
[68] A. Yazdinejad, A. Dehghantanha, R.M. Parizi, G. Srivastava, H. Karimipour, Secure intelligent fuzzy blockchain framework: effective threat detection in iot networks, Comput. Ind. 144 (2023) 103801.
[69] A. Yazdinejad, A. Dehghantanha, G. Srivastava, AP2FL: Auditable privacy-preserving federated learning framework for electronics in healthcare, IEEE Trans. Consum. Electron. 70 (2023) 2527–2535.
[70] A. Yazdinejad, A. Dehghantanha, G. Srivastava, H. Karimipour, R.M. Parizi, Hybrid privacy preserving federated learning against irregular users in next-generation Internet of Things, J. Syst. Architect. 148 (2024) 103088.