Hindawi
Security and Communication Networks
Volume 2022, Article ID 9377818, 12 pages
https://doi.org/10.1155/2022/9377818

Research Article

A Blockchain-Based Secure Radio Frequency Identification Ownership Transfer Protocol

M. Vijayalakshmi,[1] **S. Mercy Shalinie**,[1] **Ming Hour Yang**,[2] **Shou-Chuan Lai**,[3] **and Jia-Ning Luo**[3,4]

[1]*Department of Computer Science and Engineering, Thiagarajar College of Engineering, India*
[2]*Department of Information and Computer Engineering, Chuan Yuan Christian University, Taiwan*
[3]*Department of Information and Telecommunications Engineering, Ming Chuan University, Taiwan*
[4]*Department of Computer Science and Information Engineering, Chung Cheng Institute of Technology, National Defense University, Taiwan*

Correspondence should be addressed to Jia-Ning Luo; deer@mail.mcu.edu.tw

Received 23 July 2021; Accepted 26 November 2021; Published 7 January 2022

Academic Editor: Anwar Ghani

Supply chain management (SCM) governance is the streamline of the IoT product life cycle from its production to delivery. Integrating blockchain with supply chain management is essential to ensure end-to-end tracking, trustiness between manufacturers and customers, fraud and counterfeit elimination, and customizing administrative costs and paperwork. This paper proposes an RFID ownership transfer protocol with the help of zk-SNARKs (Zero Knowledge-Succinct Noninteractive Arguments of Knowledge) using Ethereum blockchain. When the owner performs RFID transfer, the transferred information will be recorded on the blockchain using smart contracts. When using a smart contract to transfer ownership on the Ethereum blockchain, because the content on the blockchain will not be tampered with, all accounts in the Ethereum can view the transfer results and verify them. The privacy of the supply chain is attained by generating the proof of product code via zk-SNARKs algorithm. This algorithm also enhances the scalability of the supply chain system by creating a trusted setup in off-chain mode.

1. Introduction

The supply chain is a global network that involves organizations, people, resources, and activities to supply products or services to the end customers. Supply chain management (SCM) governance is the streamline of the product life cycle [1]. Supply chain management refers to the process when products are transferred from manufacturers to retailers, and finally consumers. Supply chain management includes commodity trading, logistics tracking, stock inventory, product traceability, and production line control. Standard SCM process flows such as product flow, information flow, financial flow, risk flow, and value flow are considered the heart of any supply chain scenario. Integration of all these flows is the critical success factor for the result of an effective supply chain [2]. The supply chain incorporates many entities such are manufacturers, distributors, retailers, and customers, who are known as actors of the SCM. Each actor plays a significant role in every phase of the product life cycle.

Even though the supply chain concept is introduced in earlier decades, now only the organizations realize the importance of it. Most of the organizations believe that an effective SCM is the only factor to extend the business relationships of an enterprise for increasing their competence in the global market [3]. This interorganizational supply structure is noticed as a virtual corporation. Globalization, proliferation, outsourcing, and information technology are considered the root causes of this modern interorganizational supply chain culture.

The efficiency of the integrated supply chain mainly depends on the information management and product traceability. However, the effective integrated supply chain means a lot of work for the organization. The more it gets globalized, the more it gets complicated as it might lack product traceability and warehouse management. Currently, most of the businesses rely on centralized databases to manage the SCM, and the internal system is monitored by a single administrator [4]. Although the advanced technologies like cloud computing and internet of things are used to access the centralized database from anywhere, it has issues with data's originality and security. There is also a possibility of cybercrimes happening in the supply chain in the form of counterfeiting to tamper the original product. Many companies advanced their tracking systems from traditional barcode to radio frequency identification (RFID) tags because of the simultaneous multiple sensing capabilities [5]. But the RFID tags are also compromised by security and privacy threats like sniffing, tracking, RFID counterfeiting, repudiation, denial of service, and replay attacks [6–8]. Therefore, to manage the tracking and warehouse system of the globalized supply chain efficiently and to mitigate the tampering attack on the products, the supply chain network has collaborated with the blockchain technology [9].

Once the product is ready to supply, the tracking system of the respective supply chain begins to trace the product based on the tagged ID (i.e., barcode, RFID, etc.). Even though this approach is well adopted, it has scalability issues and produces various bottlenecks [6]. The blockchain technology is integrated to handle the main challenges of globalized supply chain remaining in the step-by-step traceability process from manufacturers to the end customers and managing the information along with products [9, 10].

Radio frequency identification (RFID) is mainly composed of an RFID tag (Tag) and a tag reader (Reader). According to the power source of RFID tags, tags can be divided into two categories: active and passive: passive tags generate electromagnetic induction by electromagnetic waves emitted by the reader, allowing the tag to generate current to transmit data to the reader; the active tag contains a power source and is usually in a hibernation state. When the reader is in the sensing range, the tag will be awakened to communicate with the reader. Because RFID tags have smaller reading restrictions than traditional barcodes, they are now used in many fields, such as medical fields, antitheft systems, transportation field, or supply chain management. You can also see the use of RFID in life, including pet chips, leisure cards, and access control cards, all of which are RFID applications.

Because radio frequency identification (RFID) tags are more convenient to read than traditional barcodes, they are widely used in supply chain management. In supply chain management, the process of product transfer from the original owner to the next owner is called ownership transfer. During the transaction, the ownership of the RFID tag will be transferred to the new owner. However, because sellers may forge or modify product information, many scholars have proposed supply chain management using blockchains. Because the blockchain has the nontamperable feature, it can ensure the integrity and correctness of product information.

Blockchain is an open, immutable distributed ledger that enables secure transactions between a network of organizations or individuals who may have trust issues with one another by using consensus algorithms. Distributed computing and cryptography are the two prerequisites of blockchain technology [11]. Distributed technology is used to handle bottleneck issues and single point of failure. Cryptography is used to ensure the immutability of a ledger to tamper proof transactions in an untrusted network. A ledger containing the information about all transactions is distributed across each peer of the network. This distributed ledger technology is adopted in many areas that include healthcare, music, internet of things (IoT), government, passports, sensors, and smart appliances to handle a large set of information.

Each transaction in the distributed ledger is validated and updated in every block using the consensus algorithm. The most famous consensus algorithms of the blockchain are proof of work (PoW), proof of stake (PoS), proof of capacity (PoC), practical byzantine fault tolerance (PBFT), and proof of elapsed time [6]. All peers or nodes in the network must agree to the consensus algorithm. In blockchain, multiple blocks try to validate the new transactions and mine the blocks to add them into it, as shown in Figure 1.

These blocks are called as miners. Whenever a new block is mined, it gets confirmed and added to the ledger. Cryptography secured hash function is used to connect the blocks in a blockchain in a tamper proof way. Immutability of the blockchain is ensured using this one-way hash function. Every block stores the hash value of the previous block in their header. This chain of hashes forms a tree structure known as hash tree or Merkle tree, as shown in Figure 2. Any change in any transactions will reflect in Merkle root.

Blockchain is split into three major types that are public, private, and consortium based on the parties involved and the node visibility of the network, as shown in Table 1. In a permissionless (public) blockchain, any node can join the network without any limitations, while the permissioned (including both private and consortium) blockchain restricts the access rights to a set of authenticated nodes only.

This paper uses the Ethereum blockchain platform to transfer ownership of supply chain management. Ethereum can run code between distributed systems, and outsiders cannot tamper with the program. These codes will be added to the blockchain, and the code cannot be altered after programming. In addition, everyone can audit the code before interacting with it. This means that anyone anywhere can launch applications that cannot be obtained offline. We call the programs that make up the application smart contracts. In most cases, they can be set to operate without human intervention.

Supply chain management mainly involves transactions between manufacturers, distributors, retailers, and consumers. Manufacturers are individuals or organizations that produce products. In the Ethereum blockchain platform, the manufacturer is responsible for registering the product, as well as for marking and scanning before delivery; the distributor is the intermediary between the manufacturer and the retailer, responsible for transferring the product, so on

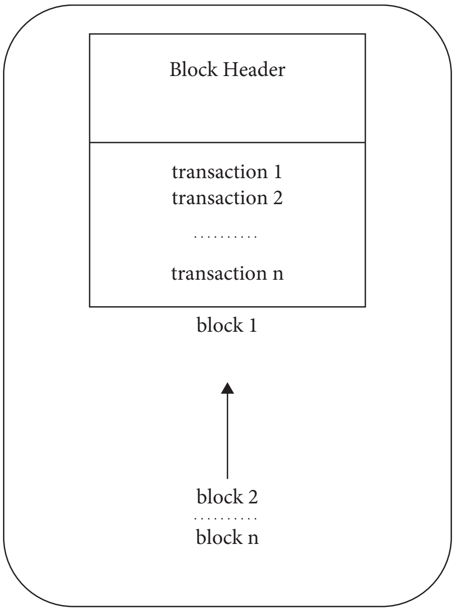

Figure 1: Structure of the distributed public ledger.

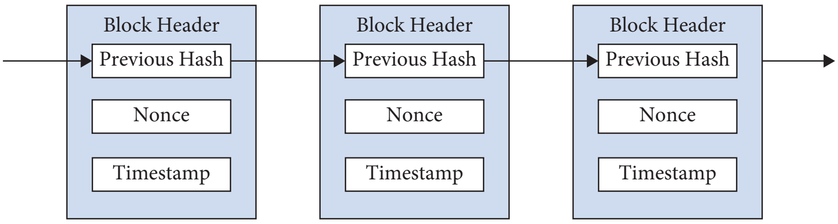

Figure 2: Structure of the hash chain.

Table 1: Three major types of blockchain.

Metrics	Public	Private	Consortium (semiprivate)
Access permissions	Open	Restricted	Restricted
Network	Fully decentralized	Partially decentralized	Partially decentralized
Platforms	Bitcoin, Ethereum, Litecoin, etc.	Hyperledger, Ripple, etc.	Bankchain, B3i, Enterprise Ethereum Alliance (EEA), etc.
Transaction validator	Anyone	Single entity	Multiple entities
Participant identity	Unknown	Known	Unknown
Required transaction time	High	Low	Low
Transaction throughput	Low	High	High
Transaction speed	Slow	Fast	Fast
Security	Achieved through consensus algorithms such as proof of work (PoW), proof of stake (PoS), proof of capacity (PoC), etc.	Achieved through predetermined nodes and access restrictions	Achieved through predetermined nodes and access restrictions
Transparency	Fully transparent across the network	Restricted to some nodes of the network	Fully transparent across the network

this platform you can get rewards from the manufacturer; the retailer is responsible for purchasing products from the manufacturer and then selling the products to consumers for profit. Finally, when consumers purchase products, they have the right to verify the source of the products and can check the logistics at any time.

We deposit a reverse hash chain on the RFID tag, and each time the ownership is transferred, the RFID tag will announce the hash value of a new reverse hash chain. The nodes on the blockchain will be able to use smart contracts to check whether the hash value of this transaction can generate the hash value of the previous transaction after the hash operation. Only when the calculated hash value is the same can the correctness of this transaction be confirmed.

In the proposed ownership transfer protocol, smart contracts are used on the Ethereum blockchain platform to realize the transfer of ownership of RFID tags. The smart contract can control the content of the contract according to the program logic. Once the smart contract is deployed, the content of the contract cannot be changed, and the content is stored in the blockchain.

2. Related Work

Numerous researches have been conducted in blockchain integrated supply chain area to enhance the product traceability, security, transparency, and reliability of the supply chain. Some of them focused on the security of the SCM to overcome counterfeiting, fraud, tracking attacks, and tampering.

Toyoda et al. [12] proposed a novel product ownership management (POMS) in 2017 that uses blockchain technology to encounter counterfeit attacks in the post-supply chain scenario using blockchain technology. Forgers cannot prove that they own products in the system, so it is difficult for counterfeiters to copy real RFID tags. The post-supply chain proposed by the author means that after retailers, goods are transferred to buyers through transactions, and buyers then resell through second-hand markets (as shown in Figure 3). Buyers can read product information from the RFID tag, which can avoid buying counterfeit products. And during the transaction, both parties can confirm the transaction through the product management contract. When one of the parties refuses the transaction, the transaction will fail. Furthermore, Chen et al. proposed a hybrid scheme to combine RFID and blockchain technologies to collect patient physiological signals in medical research [13].

Ownership of the product is verified by the novel "proof of possession of products" concept admired by bitcoin's "proof of possession of balance" [14]. The proposed work is implemented on the Ethereum platform [15] and evaluated in TestRPC [16] environment. However, the genuine product without proper ownership information is also considered as counterfeit in this system. Moreover, the fixed electronic product code (EPC) of a product might also lead to tracking attack.

Qijun Lin et al. [17] proposed a food safety traceability system for a food supply chain scenario with the help of blockchain and EPC information. Off-chain and on-chain concept is used to manage information to increase the performance of the system. The proposed system is implemented on Ethereum platform via smart contracts. The designed smart contracts need to get optimized to increase the efficiency of the system. Pinchen Cui et al. [18] designed a blockchain-based supply chain framework to enhance the traceability of electronic parts or devices by providing a unique ID for each product. This framework is implemented on the Hyperledger Fabric platform. Even though this framework is implemented in permissioned blockchain, it could not protect over illegitimate manufacturer registration issues. Federico Matteo Bencic et al. [19] invented a novel distributed ledger- (DL-) tags (smart tag) approach to verify the authenticity of the products while ensuring the privacy of stakeholders and customers. This approach is implemented on Ethereum and evaluated in a real-time use case scenario, tag it wine (TIW). However, this smart tag approach requires high cost and is computationally expensive. Shangping Wang et al. [6] proposed a decentralized product traceability system based on blockchain technology which is implemented using Ethereum.

Smart contracts are designed to process product registration, transferring, and tracking in a supply chain. An event-response mechanism is designed to mitigate the man-in-the-middle attack. However, this approach cannot prevent the tracking attack. Michail Sidorov et al. [20] designed an ultra-lightweight mutual authentication RFID protocol that works together with a decentralized database to create a secure blockchain enabled SCM. The protocol mainly involves RFID tags, tag readers, and supply chain nodes. The supply chain nodes are composed of manufacturers, distributors, retailers, and consumers. The proposed protocol is coded using High Level Protocol Specification Language (HLPSL) [21] to be formally verified using a broadly accepted formal verification tool, that is, AVISPA [22]. This protocol mainly protects the communication security between the reader and the tag. When the goods are handed over to the buyer, the buyer can view the record of the traceable product, and the buyer can write its data into the blockchain after being authenticated. However, this protocol is suitable only for the supply chain with permissioned blockchain.

Sidorov et al. [20] proposed an ultra-lightweight mutual authentication RFID protocol in 2019, which is applied to supply chain management and executed on a blockchain platform. The agreement mainly involves RFID tags, tag readers, and supply chain nodes. The supply chain nodes are composed of manufacturers, distributors, retailers, and consumers.

Leonardo Aniello et al. [23] proposed a secure SCM using blockchain and physically unclonable functions (PUF) [24]. PUF are used to generate the unique unclonable tamper proof IDs. This approach is implemented on the consortium blockchain platform. However, the proposed tracking system cannot fully prevent forgery, Byzantine attack, and privacy issues. Counterfeiters use these nooks and holes of the supply chain for their own desires. Therefore, a robust and secure supply chain management is a much needed one in the current scenario. D. Islam assumes that the RFID tag is PUF-enabled [25].

The aim of the proposed system is to develop a secure supply chain management to enhance the product traceability, anticounterfeiting, and information management with the help of blockchain technology. This proof-of-

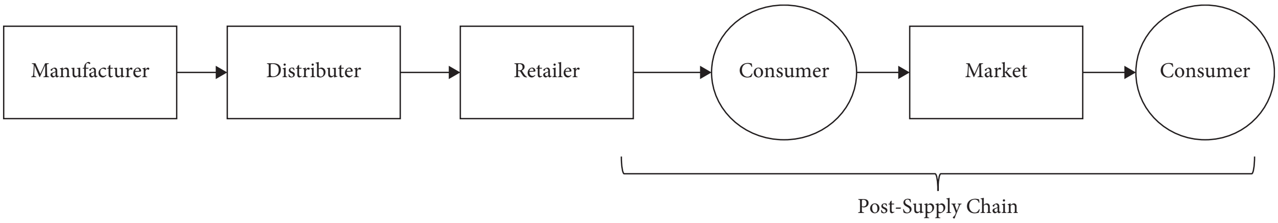

Figure 3: Post-supply chain.

concept experiment is implemented on the Ethereum platform. Smart contracts are designed to enable the transactions between the entities of a supply chain. The deployed transactions are then transferred via MetaMask crypto wallet. The privacy of the supply chain is attained by generating the proof of product code via zk-SNARKs algorithm [26]. This algorithm also enhances the scalability of the supply chain system by creating a trusted setup in off-chain mode.

3. System Model

This paper uses the Ethereum blockchain platform to transfer ownership of supply chain management. The blockchain integrated supply chain model manages the transactions between the actors of SCM, including the manufacturers, distributors, retailers, and consumers. Each of these actors plays a significant role in every phase of the SCM:

(i) Manufacturers: a manufacturer is an individual or an organization which produces a complete product from the raw materials to make profit by delivering them to the end customers. The manufacturers are those responsible for registering the product and for tagging and scanning before supplying it. In the Ethereum blockchain platform, the manufacturer is responsible for registering the product, as well as for marking and scanning before delivery

(ii) Distributors: distributors act as a mediator between the manufacturer and the retailer. They just simply transfer the products between them. For doing that job, they get incentives from the manufacturers. Finally, when consumers purchase products, they have the right to verify the source of the products and can check the logistics.

(iii) Retailers: retailers act as a gatekeeper between the manufacturer and customer, and they play a significant role in production and consumption [27]. Retailers purchase the product from the producers and sell it to the customers with some increment in the product cost to make profit.

(iv) Customers: customers are the final entity of the supply chain. A customer is an individual who buys the product and uses it. When consumers purchase products, they have the right to validate the originality of the product and to track the product during the shipment period.

Figure 4 depicts the overview of a blockchain enabled supply chain. Ethereum blockchain platform is used to design the smart contracts for managing the transactions of a supply chain scenario [28]. Each smart contract is deployed on the blockchain and is immutable.

3.1. Integration of Blockchain and RFID Tags.

3.1. Integration of Blockchain and RFID Tags. We use reverse hash chain technology in the protocol. The manufacturer will randomly create an initial value w_n, as a seed of a hash chain, and write it into the RFID tag, and then the tag will be able to sequentially calculate the values w_{n-1}, w_{n-2}, w_{n-3}, . . ., w_2, w_1, w_0 of the hash chain.

$$\begin{aligned}
w_{n-1} &= h(w_n),\\
w_{n-2} &= h(w_{n-1}) = h^2(w_n),\\
w_{n-3} &= h(w_{n-2}) = h^2(w_{n-1}) = h^3(w_n),\\
&\cdots\\
w_2 &= h(w_3) = h^2(w_4) = h^3(w_5) = \ldots = h^{n-2}(w_n),\\
w_1 &= h(w_2) = h^2(w_3) = h^3(w_4) = \ldots = h^{n-1}(w_n),\\
w_0 &= h(w_1) = h^2(w_2) = h^3(w_3) = \ldots = h^n(w_n).
\end{aligned} \tag{1}$$

From the equations, we can calculate $w_0 = h^n(w_n)$, where n is the maximum count of transfers for each RFID tag.

When the smart contract is deployed, the manufacturer will write the hash value w_0 and the maximum number of transfers n corresponding to the RFID tag. When the tag is transferred, the tag will calculate the next w (e.g., w_i) and write it into the smart contract. At this point, the nodes of the blockchain will be able to check:

$$w_i = h(w_{i+1}) = h(h(w_{i+2})) = h^2(w_{i+2}) = \ldots = h^{(n-i)}(w_n). \tag{2}$$

If the calculation result is correct, it means that the ownership transfer of the RFID tag is legal, as shown in Figure 5.

3.2. Ownership Transfer. In the system initialization, the tag is owned by the manufacturer. Then, the ownership of the tag will be transferred from the original owner to the new owner (the manufacturer, the distributor, the retailer, and the consumer).

The procedure of ownership transfer is shown in Figure 6, which contains 11 steps:

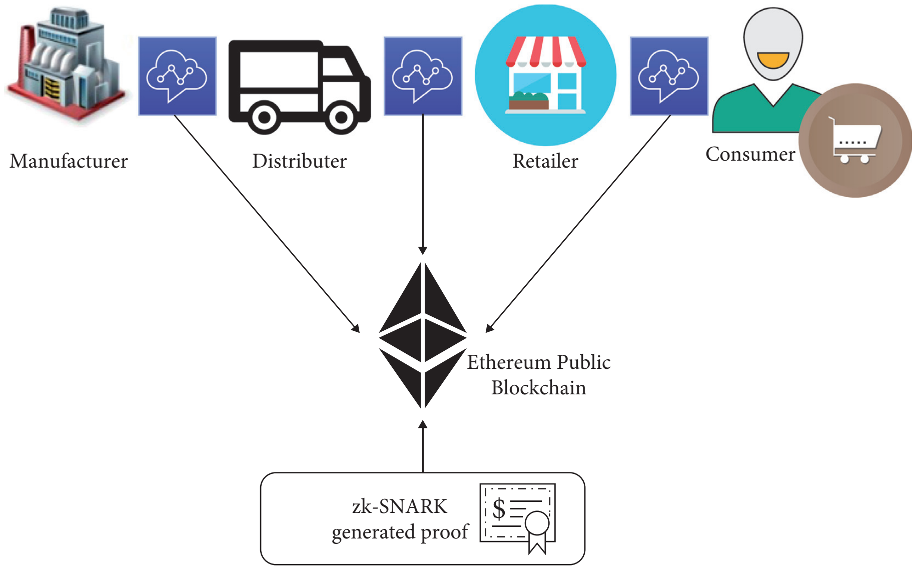

Figure 4: The overview of blockchain enabled supply chain.

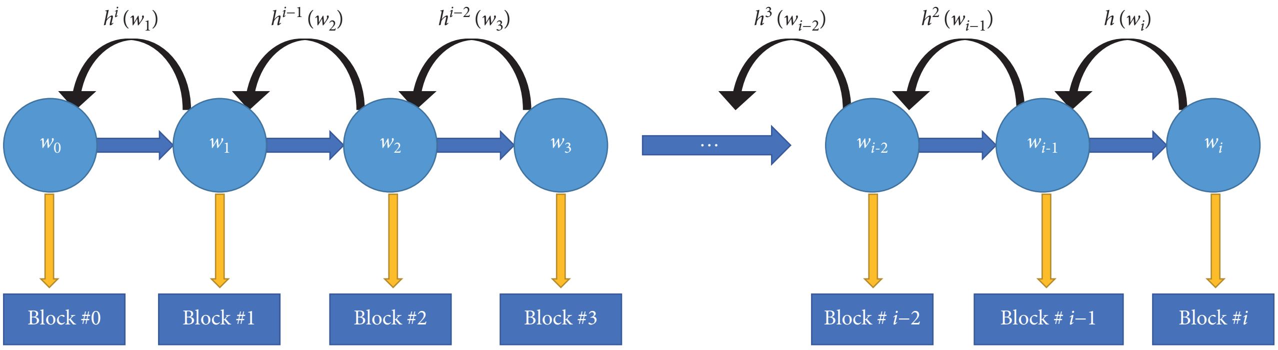

Figure 5: The reverse hash chain structure.

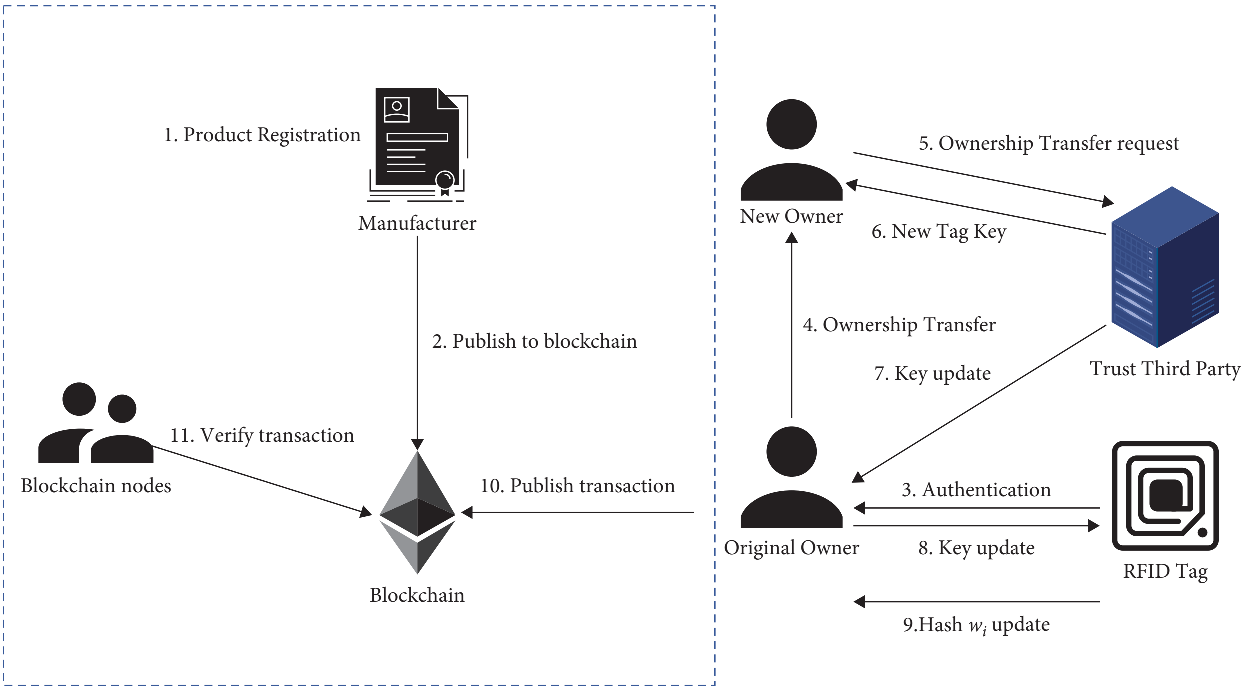

Figure 6: Integration of RFID ownership transfer with blockchain.

(1) The manufacturer creates an RFID tag with an initial seed w_n and stores its product information in the smart contract. The product information includes the initial owner, the identifier of a specific RFID tag, and the seed value.

(2) When the smart contract is released to the Ethereum blockchain platform, the product information will be written into the blockchain, and all nodes of the blockchain network can view the content. After that, the smart contract processes the transfer of RFID ownership transfer procedure.

(3) When the original owner wants to transfer the ownership of an RFID tag to the new owner, the original owner will send an ownership transfer request to the RFID tag. After the tag receives the request, it will return its information to the original owner, and the original owner can obtain the tag's identity from the server through the tag information.

(4) The tag's information obtained by the original owner will be sent to the new owner, and after receiving the tag's information, the new owner can request the transfer of ownership from the trusted third party.

(5) When the new owner requests the transfer of the tag's ownership from the TTP, the tag authentication information will be sent to TTP. Then, TTP will confirm whether the label requested for ownership transfer is the same as the identifier on the authentication message.

(6) After TTP confirms the correctness of the tag, TTP will generate a new tag key and pass the key to the new owner.

(7) TTP will update the key information and deliver it to the original owner. The original owner can update the tag's key.

(8) The original owner transfers the ownership by updating the management key of the tag.

(9) When the RFID tag completes the ownership transfer, its stored hash value w_i will be updated to $w_{(i+1)}$, and it will be published to the Ethereum blockchain network.

(10) The original owner will post transaction information to Ethereum. Ethereum will send a broadcast to each node to verify the transaction.

(11) The node will verify whether $w_0 = h(w_1)$. If it is, it is judged to be a legal transaction.

3.3. System Initialization.

3.3. System Initialization. In this proposed system, the smart contract is designed to develop a blockchain integrated secure supply chain scenario. Smart contracts are agreements written in solidity language, which stipulate transactions between entities who agree to interact with one another [29]. These smart contracts are immutable once they get deployed. The values and rights which are managed by the smart contract are stored in the blockchain. This experimental system is developed based on the "proof of ownership" concept.

Notations used in this paper are listed in Table 2.

In Step 1 of the proposed protocol is the system initialization, as shown in Table 3. The manufacture first creates a new smart contract. When the smart contract is initialized, the manufacturer writes a serial number of each RFID tag (**TagID**) and the seed value of the reverse hash chain w_0 into the contract using the smart contract constructor **constructor()**.

This function **EnrollProduct** registers the initial owner of the product and the identifier of the RFID tag, as shown in Table 4. The product will have an RFID tag attached. When the contract is released to the Ethereum blockchain platform, the product information will be written into the blockchain.

In Step 2, when the manufacturer delivers the product to the distributors, the manufacturer calls the **Add_ownership()** function to assign the ownership to the distributors that own the product, as shown in Table 5. AddOwnership() contract checks the authenticity of the requester node. If it is valid, then the ownership request is committed and the address of the node is added to the product code, otherwise rejecting the request.

3.4. Ownership Transfer. The transfer of ownership is divided into two parts. The first is the transfer of ownership of RFID tags. The old owner needs to use a reader to verify with the RFID tag before the ownership can be transferred. After the RFID transfer is completed, the ownership transfer on the blockchain is performed.

In Step 3, when the original owner wants to transfer the ownership of the RFID tag to the new owner, the original owner will send an ownership transfer request to the RFID tag. After the tag receives the request message, it will return its information to the original owner, and the original owner can verify the identity of the tag with the server. We use the protocol proposed in [30] to improve the part of RFID ownership transfer. The detailed transmission message is shown in Figure 7.

In Figure 7, when the original owner issues an ownership transfer request to the tag, the tag first generates two messages M_1 and M_2. M_1 contains a nonce and the tag's ID, which is protected by the shared key between the tag and the trusted third party (the manufacturer), and M_2 contains the tag's ID and the message M_1, which is protected by the shared key between the tag and the server of the original owner, DID^i. The tag transfers the message M_2 back to the original owner. When the original owner receives the message M_2, it generates a new message M_3 that contains the new owner's server DID^j, the new owner's ID RID_2^j, and the message M_2. The message M_3 will be transferred to the original server.

In Step 4, the original owner's server will first verify the owner when obtaining the RFID tag information. After the server confirms the identity of the owner of the RFID tag (message M_3), the server sends the tag information to the

Table 2: Notations.

Symbol	Description
TID	Tag's identifier
RID	Tag's current owner (reader)
Seed	Tag's seed value
DID^i	Server that contains tag i's information
TTP	The trusted third party
TC	Counters that indicate tag's maximum transfer count
LE	Light weight symmetric encryption and decryption function
SK_i	Shared secret
K_j^i	Session key between i and j
r_i	Random numbers
M_i	Messages transferred between entities

Table 3: System initialization function.

```
contract RFIDTag{
        address public creator;
        address public owner;
        byte32 public seed;
        uint public TagID;
        constructor() public
              creator = msg.sender;
              owner = creator;
        }
}
```

Table 4: EnrollProduct function.

```
function EnrollProduct(uint tid, uint seed) {
        require(msg.sender = = creator);
        Seed = seed;
        TagID = tid;
}
```

Table 5: Add ownership function.

```
function Addownership(uint tid, address receiver) public {
        require(msg.sender = = creator);
        if (TagID = = tid)
              owner = receiver;
        else
              revert ("the TagID is incorrect!");
   }
```

new owner and the new owner's server (message M_4). Only when the new owner server receives the tag's information and confirms that the new owner belongs to this server, the server can request the transfer of the tag's ownership from the trusted third party. If the verification is failed, a garbage message M_9 containing only a random number r_3 will be sent to the new server. In this way, the attacker has no way of knowing whether the transaction was successful. The process is shown in Figure 8.

In Steps 5-6, the server of the new owner DID^j sends the transfer request M_5 to the trusted third party (the manufacturer). The TTP verifies whether the tag requesting the ownership transfer and the tag authentication message are the same tag. After that, the TTP will create a new tag management key Kx_1^j and send it to the new server via message M_6, as shown in Figure 9.

In Step 7, TTP will update the key (message M_{10}) and encrypt it with message M_7 to send to the original owner's server DID_i. The original owner server will first confirm whether the original owner of the RFID tag in the message M_7 is correct. Then, the original owner's server will generate a message M_8 to the original owner RID_1^i. If M_7 is incorrect, another garbage message M_8 will be generated by encrypting a random number r_3, The process is shown in Figure 10.

In Steps 8-9, the original owner sends the key update message M_9 to the RFID tag. The tag updates its management key Kx_1^i for ownership transfer. When the RFID tag completes the ownership transfer process, its stored hash value w_i will be updated to $w_{(i+1)}$ and will be sent to the original owner, as shown in Figure 11.

In Step 10, as in Table 6, ChangeOwnership() smart contract verifies both parties authentication. If both the seller and buyer are valid and seller possesses the ownership, then the ownership is assigned to the new owner. And the old ownership is removed from the product code.

The original owner will use the emit Transfer() function to publish transaction messages to Ethereum. Ethereum will send a broadcast to each node to make the node verify the transaction.

In Step 11, after each node receives the notification, it will execute OwnershipTransfer (address receiver, uint amount, uint challenge) to verify whether Seed = hash(challenge). If it is, it is judged to be a legal transaction.

Our method can integrate the ownership transfer process of the RFID with the blockchain. The blockchain can record every transaction of the ownership transfer. Because of the characteristics of the blockchain, these transaction records are not easy to be tampered with, and the success of the transaction can be proved through node verification. In addition to the transfer of ownership of the blockchain, RFID tags will also transfer ownership. After the original owner transfers the ownership of the RFID tag to the new owner, the original owner will not be able to read the RFID tag, and this RFID ownership transfer agreement can satisfy the forward and backward security.

4. Experimental Evaluation

The proof-of-concept experimental system is implemented on the Ethereum remix platform. Ethereum is a public blockchain integrated development environment (IDE) used to develop various decentralized applications [29]. The cryptocurrency needed to run the application is called ether in Ethereum. Smart contracts written in solidity language are run on the Ethereum virtual machine (EVM). Every transaction in the Ethereum has a gas limit. When the gas limit expires, the transaction will be automatically aborted. The developed approach is evaluated using MetaMask's Rinkeby test network [31]. MetaMask is a crypto wallet that provides ether to blockchain apps.

Table 7 shows the utilized amount of gas and ethers by both smart contracts.

Initial ownership contract requires 0.00027 ethers, and the used gas limit is 27239. The gas price offered for this transaction is 1 Gwei. Similarly, change ownership requires

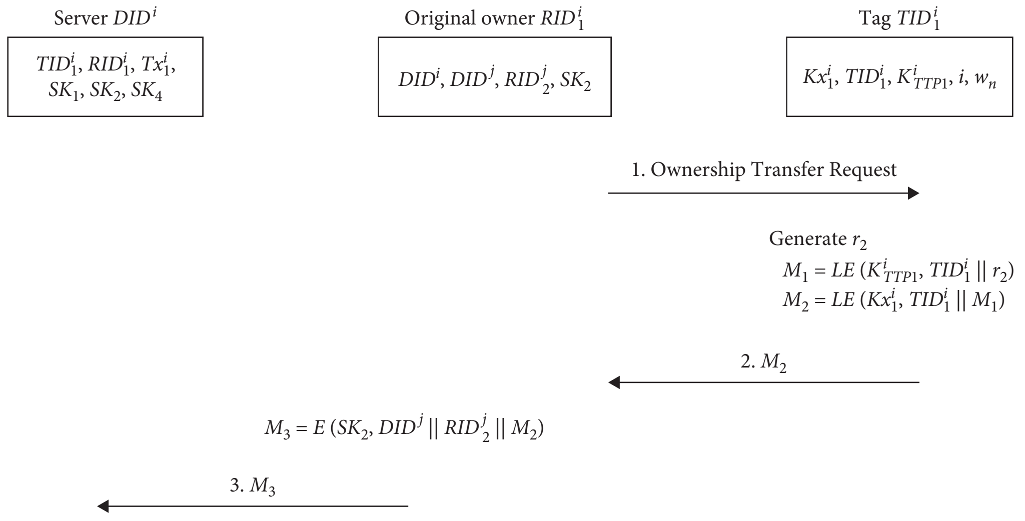
Server DID^i
$TID_1^i, RID_1^i, Tx_1^i, SK_1, SK_2, SK_4$
Original owner RID_1^i
$DID^i, DID^j, RID_2^j, SK_2$
Tag TID_1^i
$Kx_1^i, TID_1^i, K_{TTP1}^i, i, w_n$
1. Ownership Transfer Request
Generate r_2
$M_1 = LE\,(K_{TTP1}^i, TID_1^i \,\|\, r_2)$
$M_2 = LE\,(Kx_1^i, TID_1^i \,\|\, M_1)$
2. M_2
$M_3 = E\,(SK_2, DID^j \,\|\, RID_2^j \,\|\, M_2)$
3. M_3

Figure 7: RFID tag authentication with the original owner.

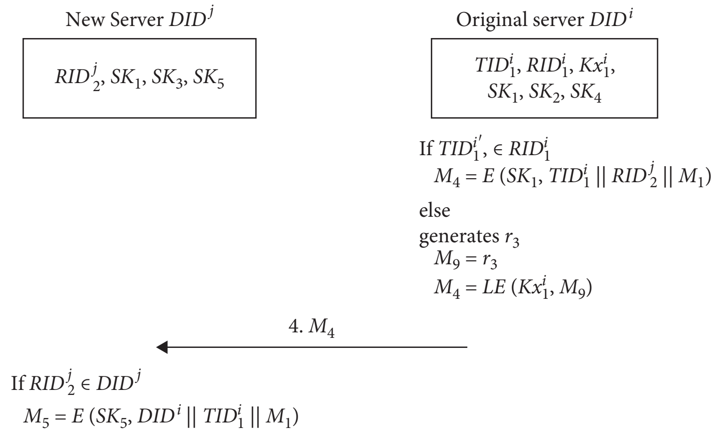
New Server DID^j
$RID_2^j, SK_1, SK_3, SK_5$
Original server DID^i
$TID_1^i, RID_1^i, Kx_1^i, SK_1, SK_2, SK_4$
If $TID_1^{i'}, \in RID_1^i$
$M_4 = E\,(SK_1, TID_1^i \,\|\, RID_2^j \,\|\, M_1)$
else
generates r_3
$M_9 = r_3$
$M_4 = LE\,(Kx_1^i, M_9)$
4. M_4
If $RID_2^j \in DID^j$
$M_5 = E\,(SK_5, DID^i \,\|\, TID_1^i \,\|\, M_1)$

Figure 8: RFID tag ownership transfer between two servers.

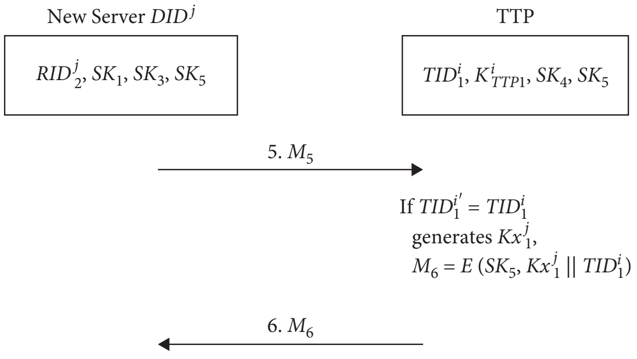
New Server DID^j
$RID_2^j, SK_1, SK_3, SK_5$
TTP
$TID_1^i, K_{TTP1}^i, SK_4, SK_5$
5. M_5
If $TID_1^{i'} = TID_1^i$
generates Kx_1^j,
$M_6 = E\,(SK_5, Kx_1^j \,\|\, TID_1^i)$
6. M_6

Figure 9: TTP verification.

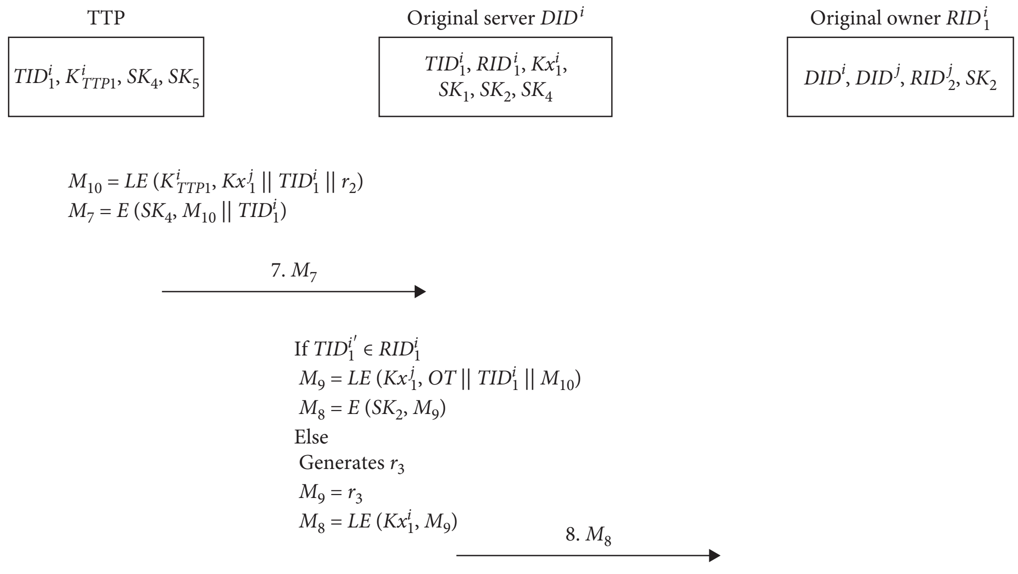
TTP
Original server DID^i
Original owner RID_1^i
$TID_1^i, K_{TTP1}^i, SK_4, SK_5$
$TID_1^i, RID_1^i, Kx_1^i, SK_1, SK_2, SK_4$
$DID^i, DID^j, RID_2^j, SK_2$
$M_{10} = LE\,(K_{TTP1}^i, Kx_1^j \,||\, TID_1^i \,||\, r_2)$
$M_7 = E\,(SK_4, M_{10} \,||\, TID_1^i)$
7. M_7
If $TID_1^{i'} \in RID_1^i$
$M_9 = LE\,(Kx_1^j, OT \,||\, TID_1^i \,||\, M_{10})$
$M_8 = E\,(SK_2, M_9)$
Else
Generates r_3
$M_9 = r_3$
$M_8 = LE\,(Kx_1^i, M_9)$
8. M_8

Figure 10: Original server and owner verification.

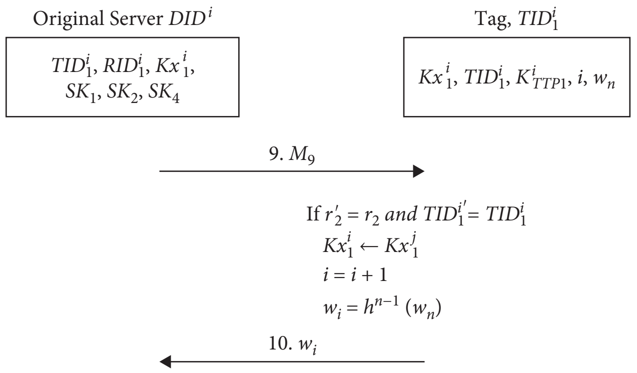
Original Server DID^i
Tag, TID_1^i
$TID_1^i, RID_1^i, Kx_1^i, SK_1, SK_2, SK_4$
$Kx_1^i, TID_1^i, K_{TTP1}^i, i, w_n$
9. M_9
If $r_2' = r_2$ and $TID_1^{i'} = TID_1^i$
$Kx_1^i \leftarrow Kx_1^j$
$i = i + 1$
$w_i = h^{n-1}\,(w_n)$
10. w_i

Figure 11: Tag’s key update and hash update.

Table 6: Change ownership function.

```
ChangeOwnership (uint tid, address receiver, uint challenge)
public {
        require amount≤owner_weights[msg.sender]);
        if (tid = = TagID && Hash(challenge) = = seed)
        {
              owner = receiver;
              Old_Seed = Seed;
              Seed = Hash(challenge);
              emit Transfer (msg.sender, receiver, TagID, seed);
        }
        else
              revert (“Failed”);
}
event Transfer (address from, address to, uint tid,
bytes32 seed);
```

Table 7: Cost evaluation of deployed smart contracts.

Contracts	Amount of gas used	Amount of ethers used
Initial ownership	26502	0.00025319
Change ownership	31486	0.00076298

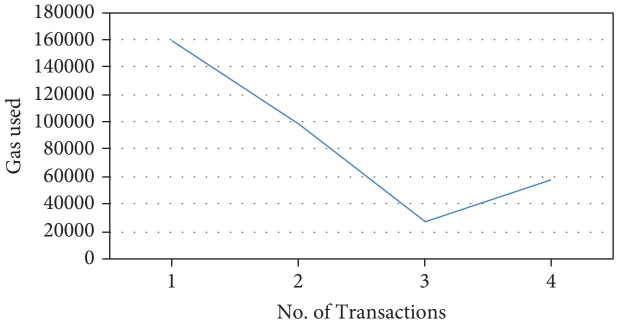

Figure 12: Transaction cost versus number of transactions.

Table 8: Attack analysis with existing approaches.

	Counterfeiting Attack prevention	Tracking Attack prevention	Tampering Attack prevention	Man-in-the-middle Attack prevention
[12]	Yes	No	Yes	Yes
[17]	No	No	Yes	No
[18]	Yes	No	Yes	Yes
[19]	Yes	No	Yes	Yes
[6]	Yes	No	Yes	Yes
Proposed design	Yes	Yes	Yes	Yes

0.000579 ethers, and the used gas limit is 57919. The gas price offered for this transaction is 2 Gwei. When comparing both contracts, change ownership has more computational time than the initial ownership contract.

Figure 12 represents the amount of gas used per transaction. The efficiency of the transaction is determined by the amount of gas used and the transaction fees (ether value).

5. Attack Analysis

Privacy and security are clearly needed for enterprise-level blockchains. The most impactful evaluation matrices, which differentiate the performance of both permissioned and permissionless blockchains, are scalability and privacy. While the public blockchain has issues with privacy, private blockchains outperform those issues by applying their distinctive characteristics such as permissioned mode of operation and fine-grained access control.

The previous researches conducted in the area of supply chain management using public blockchain are failed to discover this issue. The aim of the proposed work revolves around the development of open access supply chain management with guaranteed privacy and security.

This approach attempts to cover the privacy issues by generating the proof of information with the help of zk-SNARKs algorithm while verifying the ownership of the entity.

Table 8 shows the major supply chain attacks, such as counterfeiting attack, tampering attack, tracking attack, and man-in-the-middle attack, which are handled by the proposed approach. The analysis of these attacks against the existing approaches is represented in the table. Our proposed system covers the tracking attack while the existing approaches [6, 12, 17, 19] using Ethereum failed to prevent that attack. And this permissionless system's performance matches the permissioned system [17] when handling the verification and tracking.

6. Conclusion

An effective supply chain implementation is an essential one for every organization. The complications of traditional supply chain management systems such as tracking and information management can be mitigated with the inclusion of blockchains. In this paper, we propose an RFID ownership transfer protocol that can integrate the ownership transfer process of the RFID on the blockchain. In this proposed system, a secure supply chain management is developed in order to enhance the product privacy, anticounterfeiting, traceability, and information management with the help of zk-SNARKs using Ethereum blockchain by means of smart contracts. The proposed approach is implemented on Ethereum blockchain and evaluated using MetaMask's Rinkeby test network.

Data Availability

The simulation results used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Taiwan Information Security Center (TWISC) and the Ministry of Science and Technology (MOST) under Grants nos. MOST 109-2221-E-130-006 and MOST 110-2221-E-033-008.

References

[1] H. Werner, *Supply Chain Management*, Springer, Berlin, Germany, 2000.

[2] C. K. H. Lee, K. L. Choy, K. M. Y. Law, and G. T. S. Ho, "Application of intelligent data management in resource allocation for effective operation of manufacturing systems," *Journal of Manufacturing Systems*, vol. 33, no. 3, pp. 412–422, 2014.

[3] S. krishna, *The Five Major Flows in Supply Chain*, 2016, https://brandalyzer.blog/2016/03/23/the-five-major-flows-in-supply-chain/.

[4] E. B. Sasson, A. Chiesa, C. Garman et al., "Zerocash: decentralized anonymous payments from bitcoin," in *Proceedings of the 2014 IEEE Symposium on Security and Privacy*, pp. 459–474, IEEE, Washington, DC, USA, May 2014.

[5] K. Domdouzis, B. Kumar, and C. Anumba, "Radio-frequency identification (rfid) applications: a brief introduction," *Advanced Engineering Informatics*, vol. 21, no. 4, pp. 350–355, 2007.

[6] S. Wang, D. Li, Y. Zhang, and J. Chen, "Smart contract-based product traceability system in the supply chain scenario," *IEEE Access*, vol. 7, pp. 115122–115133, 2019.

[7] M. H. Yang and J.-N. Luo, "Authentication protocol in mobile rfid network," in *Proceedings of the 2009 Fourth International Conference on Systems*, pp. 108–113, IEEE, Washington, DC, USA, September 2009.

[8] J.-N. Luo and M.-H. Yang, "An efficient offline delegation protocol in mobile rfid environment," *Journal of Networks*, vol. 9, no. 5, p. 1114, 2014.

[9] M. M. Queiroz, R. Telles, and S. H. Bonilla, "Blockchain and supply chain management integration: a systematic review of the literature," *Supply Chain Management: International Journal*, vol. 25, no. 1, 2019.

[10] H. Treiblmaier, "The impact of the blockchain on the supply chain: a theory-based research framework and a call for action," *Supply Chain Management: International Journal*, vol. 23, no. 6, 2018.

[11] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, and C. Yang, "The blockchain as a decentralized security framework [future directions]," *IEEE Consumer Electronics Magazine*, vol. 7, no. 2, pp. 18–21, 2018.

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

[13] X. Chen, H. Zhu, D. Geng, W. Liu, R. Yang, and S. Li, "Merging rfid and blockchain technologies to accelerate big data medical research based on physiological signals," *Journal of healthcare engineering*, vol. 2020, Article ID 2452683, 2020.

[14] M. Crosby, P. Pattanayak, S. Verma, V. Kalyanaraman et al.,et al. "Blockchain technology: beyond bitcoin," *Applied Innovation*, vol. 2, no. 6-10, p. 71, 2016.

[15] C. Dannen, *Introducing Ethereum and Solidity*, vol. 1, Springer, Berlin, Germany, 2017.

[16] N. library, *Nethereum Documentation - Test Rpc Configuration and Usage*, https://docs.nethereum.com/en/latest/ethereum-and-clients/test-rpc/, 2020.

[17] Q. Lin, H. Wang, X. Pei, and J. Wang, "Food safety traceability system based on blockchain and epcis," *IEEE Access*, vol. 7, pp. 20698–20707, 2019.

[18] P. Cui, J. Dixon, U. Guin, and D. Dimase, "A blockchain-based framework for supply chain provenance," *IEEE Access*, vol. 7, pp. 157113–157125, 2019.

[19] F. M. Bencic, P. Skocir, and I. P. Žarko, "Dl-tags: dlt and smart tags for decentralized, privacy-preserving, and verifiable supply chain management," *IEEE access*, vol. 7, pp. 46198–46209, 2019.

[20] M. Sidorov, M. T. Ong, R. V. Sridharan, J. Nakamura, R. Ohmura, and J. H. Khor, "Ultralightweight mutual authentication rfid protocol for blockchain enabled supply chains," *IEEE Access*, vol. 7, pp. 7273–7285, 2019.

[21] D. Von Oheimb, "The high-level protocol specification language hlpsl developed in the eu project avispa," in *Proceedings of the APPSEM 2005 workshop*, pp. 1–17, Frauenchiemsee, Germany, September 2005.

[22] A. Armando, D. Basin, Y. Boichut et al., "The avispa tool for the automated validation of internet security protocols and applications," *Computer Aided Verification*, Springer, in *Proceedings of the International conference on computer aided verification*, pp. 281–285, July 2005.

[23] L. Aniello, B. Halak, P. Chai, R. Dhall, M. Mihalea, and A. Wilczynski, "Towards a supply chain management system for counterfeit mitigation using blockchain and puf," *arXiv preprint arXiv:1908.09585*, 2019.

[24] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: a tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, 2014.

[25] M. D. Islam, *Integrating Blockchain into Supply Chain Safeguarded by Puf-Enabled Rfid*, Ph.D. dissertation, University of Nevada, Reno, 2021.

[26] A. M. Pinto, "An introduction to the use of zk-snarks in blockchains," in *Mathematical Research for Blockchain Economy*, pp. 233–249, Springer, Berlin, Germany, 2020.

[27] T. D. E. P. Agency, *The Role of Retailers in the Transition towards Sustainable Consumption and Production*, https://eng.mst.dk/sustainability/sustainable-consumption-and-production/green-nordic-retail/the-role-of-retailers/, 2020.

[28] G. A. Oliva, A. E. Hassan, and Z. M. J. Jiang, "An exploratory study of smart contracts in the ethereum blockchain platform," *Empirical Software Engineering*, vol. 25, no. 2, pp. 1–41, 2020.

[29] M. Mukhopadhyay, *Ethereum Smart Contract Development: Build Blockchain-Based Decentralized Applications Using Solidity*, Packt Publishing Ltd, Birmingham, United Kingdom, 2018.

[30] M. H. Yang, "Across-authority lightweight ownership transfer protocol," *Electronic Commerce Research and Applications*, vol. 10, no. 4, pp. 375–383, 2011.

[31] MetaMask, *Metamask's Developer Documentation*, https://metamask.io/, 2020.