Design and Evaluation of a Privacy-preserving Supply Chain System Based on Public Permissionless Blockchain

Takio Uesugi
Graduate School of Information Science and Technology, Osaka University
Suita, Osaka, 565-0871 Japan
t-uesugi@ist.osaka-u.ac.jp

Yoshinobu Shijo
Graduate School of Information Science and Technology, Osaka University
Suita, Osaka, 565-0871 Japan
y-shijo@ist.osaka-u.ac.jp

Masayuki Murata
Graduate School of Information Science and Technology, Osaka University
Suita, Osaka, 565-0871 Japan
murata@ist.osaka-u.ac.jp

ABSTRACT

Securing the traceability of products in a supply chain is an urgent issue. Recently, supply-chain systems that use a blockchain have been proposed. In these systems, the blockchain is used as a common database shared among supply chain parties to secure the integrity and reliability of distribution information such as ownership transfer records. These systems thus secure a high level of traceability in the supply chain. Considering future scalability of supply chains, public permissionless blockchain (PPBC) is a promising approach. In this approach, however, distribution information that should be kept private is made public since the information recorded in PPBC can be read by anyone. We therefore propose a method for preserving privacy while securing traceability in a supply chain system using PPBC. The proposed method preserves privacy by concealing distribution information via encryption. In addition, the proposed method ensures distribution among legitimate supply chain parties while concealing their blockchain addresses by using zero-knowledge proofs.We implement the proposed method on Ethereum smart contracts and verify the system behavior. The results show that the proposed method works as expected, and that system usage cost per distribution party is at most 2.2×10^6 *gas* units in terms of blockchain transaction fees.

CCS Concepts

•**Information systems** → **Process control systems;**
•**Security and privacy** → **Privacy protections;**

Keywords

Public blockchain; Privacy-preserving technology; Zero-knowledge proof; Supply chain; Traceability; Ethereum

1. INTRODUCTION

In recent years, distribution forms and relationships in the supply chain have become larger and more complex. In terms of distribution forms, in addition to primary distribution from manufacturers to consumers, secondary markets are also growing significantly, because it has become easier for individuals to conduct transactions using free-market apps and other tools. In terms of distribution relationships, various parties have newly joined the supply chain due to increasing globalization and the emergence of new business types. Against this background, information management in the supply chain has become increasingly difficult. As a provisional solution, each party has managed information by building its own database. This has led to information silos, however, resulting in serious problems, particularly with respect to verifying product origins and securing traceability. For example, the Organisation for Economic Co-operation and Development reported that counterfeit products in international trade totaled 509 billion USD in 2016, up from 461 billion USD in 2013 [1]. This could be due to the increasing difficulty in verifying the correctness of product manufacturers. Furthermore, supply chain complexity makes it difficult to track ingredients contaminated with *Escherichia coli*, resulting in an outbreak in the Chipotle Mexican Grill restaurant chain in 2015 [2].

A possible remedy to these problems is to unitarily manage distribution information among multiple parties. In order to realize this solution, supply chain systems based on blockchain have been proposed [3–5]. These systems record product distribution information, or ownership transfer records, in a blockchain. Blockchain is a decentralized time-series ledger. Blockchain network nodes independently verify and update data based on a common logic, then store that data. Invalid data are automatically detected and removed by mutual comparison of stored data, making them tamper-resistant and highly available. Therefore, distribution information in the blockchain cannot be tampered with or lost due to system failure. In addition, by using a customizable common logic called "smart contracts," it is possible to set appropriate conditions for registration and modification of distribution information, thereby preventing the storage of unauthorized distribution information.

Distribution forms and relationships in the supply chain are likely to become increasingly larger and complex. Considering the future scalability of such supply chains, it is desirable to design a supply chain system that allows anyone to freely update and browse the data. Such systems allow

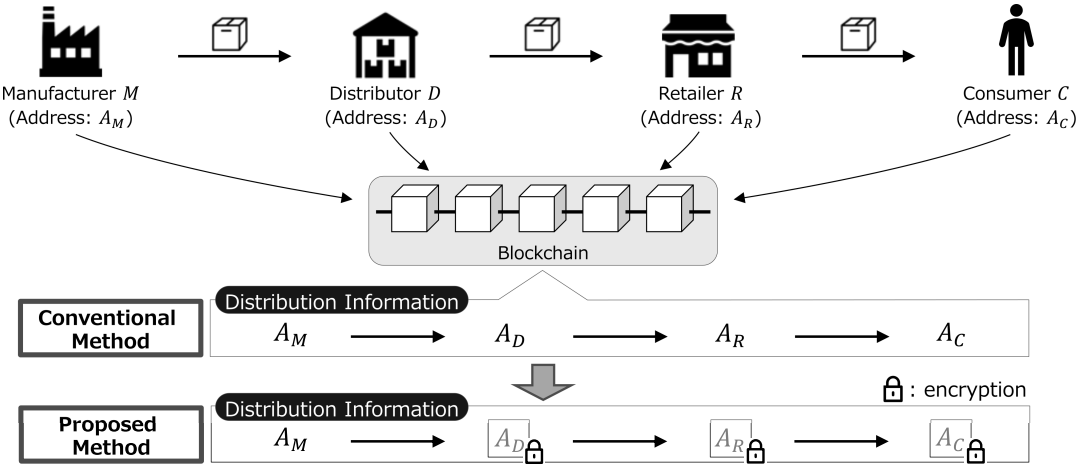

Figure 1: Overviews of conventional and proposed methods.

one to freely register distribution relationships, which does not prevent new entrants from establishing new distribution relationships. Furthermore, the system can be applied to various new distribution forms that may appear in the future. It can also provide added product value by allowing general consumers to confirm distribution information in the supply chain. For example, consumers can confirm product safety by confirming the origin of agricultural products or the manufacturer of purchased products. Blockchain can be roughly classified as public or private in terms of data reference permissions, and as permissionless or permissioned in terms of data update permissions [6]. A blockchain in which anyone can browse stored data and record new data is called a public permissionless blockchain. Such public permissionless blockchains are most appropriate for supply chains.

However, the use of public permissionless blockchain presents new challenges, particularly regarding data privacy [7]. In a public permissionless blockchain, all blockchain data are made public, even information that should be kept secret, which can be problematic in supply-chain systems. Businesses invest significant costs when investigating business partners and establishing distribution relationships to reduce purchase prices and achieve rapid distribution. Publicizing this information threatens competitive advantages, as competitors become able to identify and establish distribution relationships at little cost. As another example, transaction information between individuals and product ownership information would be identifiable in secondary markets. Such information should not be made public. Therefore, privacy needs to be preserved for distribution information. As a concrete case study, Coke One North America Services, which provides the IT platform for Coca-Cola's bottling business in North America, has been testing bottling transactions using a public permissionless blockchain to build extensive distribution relationships [8]. However, while the method they applied preserved the privacy of transaction information, it was unable to secure traceability throughout the supply chain. (See Section 2 for further details.)

In this paper, we propose a method for securing traceability and preserving privacy in a supply chain system based on public permissionless blockchain. The contributions of this paper are as follows:

- We propose a method that can secure the traceability of product distribution and preserve the privacy of distribution information in a public permissionless blockchain-based supply chain system. Privacy preservation is achieved by encryption of distribution information and a zero-knowledge proof. Figure 1 shows a comparison of the supply chain system under the conventional and proposed methods.

- We analyze attackability of the proposed method for traceability and privacy preservation and verify that it is capable of tracking distribution and preserving the privacy of distribution information against most of the attacks.

- We implement the proposed method and show that the proposed method works as expected in scenarios that verify its operation.

- We calculate the system cost incurred by operating the proposed method. Based on the results, we show a use case where the proposed method can be applied.

The remainder of this paper is structured as follows. Section 2 introduces related work. The system and privacy model of the proposed method are described in Section 3. Section 4 introduces the proposed method, which is verified in Section 5 in terms of privacy and traceability. Section 6 describes the environment for evaluating the proposed method and presents the operation results. That section also evaluates transaction fees and presents a use case of the proposed method. Lastly, our conclusions and future work are presented in Section 7.

Note that in this paper, the word "blockchain" with no supplementary explanation indicates a public permissionless blockchain.

2. RELATED WORK

Several blockchain-based systems have been proposed to improve the traceability of products in supply chains.

Hackius et al. [9] investigated the potential application of blockchain to logistics and supply chain management and introduced several use cases. Kshetri [10] provided case

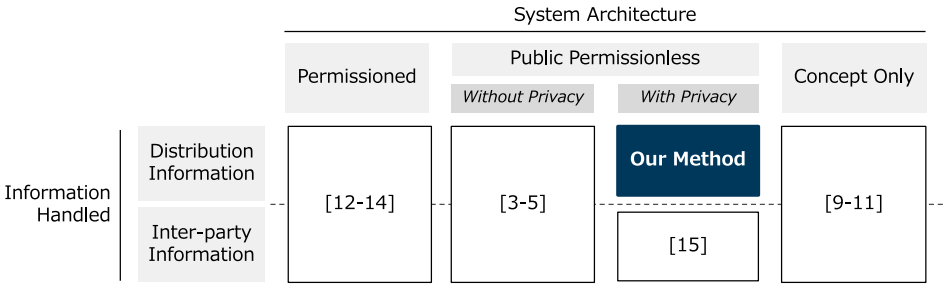

Figure 2: Positioning of the proposed method and related work.

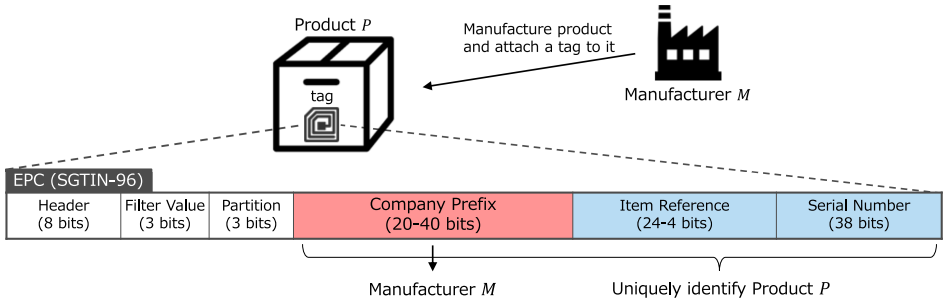

Figure 3: Information recorded on a tag attached to the product.

studies of blockchain-based supply chain systems and discussed their benefits and challenges. Tian [11] proposed a blockchain-based supply chain traceability system for agricultural products in China with the aim of securing the safety of agricultural products. However, they only proposed the concept of a blockchain-based supply chain system, without referring to specific methods or implementations.

Some researchers have proposed supply chain systems based on permissioned blockchain. Sinclair et al. [12] aimed to secure the traceability of medicines, while IBM Food Trust [13] aimed to secure the traceability of food products. Maouchi et al. [14] proposed a method for guaranteeing privacy by using permissioned blockchain to set appropriate permissions for browsing data. However, these systems limited blockchain participation, making it difficult to promote the widespread use of supply chain systems and hindering the uniform management of distribution information. As a result, it is not possible to secure product traceability. In addition, it is practically impossible to apply the system to transactions between general consumers because it is difficult for them to participate in the system.

Recently, supply chain systems based on public permissionless blockchain have been proposed. Toyoda et al. [3] proposed a method for using blockchain to manage product ownership transfers to prevent the distribution of counterfeit products in the post supply chain. Kim et al. [4] proposed a method for tracking products from the materials stage through repeated consumption and production of traceable resource units. Huang et al. [5] proposed a method for applying off-chain technologies to food-supply chains, which feature high-frequency distribution. These supply chain systems are expected to become widely used because anyone can freely participate in the supply chain and browse information. This promotes uniform management of distribution information and contributes to securing product traceability. However, none of these methods consider information privacy, so highly private information, such as transaction and ownership information, will be widely disclosed.

The Baseline Protocol [15] aims at addressing privacy issues in supply chain systems based on a public permissionless blockchain by building a database that can be commonly used by trading partners without building mutual trust. For that purpose, it stores company and transaction information in the form of zero-knowledge proofs on a public permissionless blockchain. This zero-knowledge proof ensures that no other party can obtain information from the proof information stored in the blockchain. Meanwhile, transacting parties can confirm the original information by verifying proof information. This solves the privacy problem between transacting parties and other participants in a public permissionless blockchain. However, it does not consider distribution information across the entire supply chain, and so cannot secure product traceability.

In this paper, we propose a method for using public permissionless blockchain to allow public participants to confirm manufacturers and to allow manufacturers to track their products in a supply chain system, while still preserving distribution information privacy. Figure 2 compares the proposed method and those in previous studies in terms of information handled and system architecture. We can see that only the proposed method is able to secure both privacy and traceability of distribution information in a supply chain system using public permissionless blockchain. This characteristic is the main novelty of the proposed method.

3. SYSTEM AND PRIVACY MODEL

This section introduces the system model and the privacy model assumed in the proposed method.

3.1 System model

The proposed method targets only the distribution of a

Table 1: Permission control of the proposed method.

		Parties			
		Manufacturer	Current owner	Next owner	Others
Processes	Enroll product	✓			
	Ship product		✓		
	Receive product			✓	

Table 2: Privacy model of the proposed method.

		Parties			
		Manufacturer	Current owner	Next owner	Others
Permissions	Browse manufacturer	✓	✓	✓	✓
	Browse current owner	✓	✓	✓	
	Browse next owner	✓	✓	✓	
	Browse ownership history	✓			

single product without modification of its form, that is, distribution of the finished product. Therefore, the proposed method does not consider assembly, disassembly, or aggregation of products.

We assumed that a tag such as an RFID or QR code is attached to the finished product. As Figure 3 shows, the tag includes an electronic product code (EPC) in the SGTIN-96 format. SGTIN-96 includes a company prefix identifying the product manufacturer and a serial number that identifies the product. The proposed method uses this EPC to manage product information. We assume that manufacturers can freely create EPCs and write them to the tags, and that EPCs are correct, that is, the manufacturer's company prefix and the product's unique serial number are correctly recorded. We also assume that EPCs attached to a product cannot be altered by replacing or tampering with the tag.

Among the supply chain information, the proposed method focuses only on information distribution, or transitions of product ownership. The initial product owner is its manufacturer, and ownership transfers as the product is shipped and received.

In the proposed method, we use a public permissionless blockchain with smart contract functionality. We assume that party identities on the blockchain are managed with a pair of private and public keys. Private keys have a one-to-one relationship with their owners and are carefully managed to prevent disclosure. Public keys can be freely obtained by third parties. Blockchain addresses are generated from a public key using a hash function or other method that uniquely defines values. Others cannot recover the private key from the public key or the blockchain address. One of the most famous blockchains satisfying these properties is Ethereum [16], which we use for the implementation in this paper.

To track products, we must obtain information about the parties in a certain way. Therefore, we assume that public keys and blockchain addresses are linked to party identities by the same mechanism as public key certificates, which provide party locations and names.

The proposed method tracks product distribution by processing enrollments and repeating shipment and receipt processes for the product. To prevent unauthorized product distribution, we control party permissions for running each process as shown in Table 1. In that table, "manufacturer" is the party that manufactured the product, "current owner" is the party that currently owns the product, and "next owner" is the party that will receive the product from its current owner. Checkmarks (✓) indicate processes that can be run by the corresponding parties.

3.2 Privacy Model

The proposed method manages four types of product distribution information: manufacturer, current owner, next owner, and ownership transition. The proposed method cryptographically manages permissions for party access to information as shown in Table 2, where checkmarks (✓) indicate permissions that the corresponding parties have. Definitions for each party are the same as in Table 1.

The proposed method allows anyone to browse product manufacturers at will, allowing them to confirm that products were manufactured by a legitimate manufacturer and preventing distribution of counterfeit products. However, only manufacturers can browse ownership histories, because the manufacturer is responsible for the product. Manufacturers can immediately identify owners to recall products or prevent their further distribution when a problem occurs. No other parties have reasonable grounds for browsing ownership histories. It is natural for product senders and recipients to want to verify each other's identity. Thus, the proposed method secures the privacy of distribution information by providing appropriate permission management for browsing that information.

4. PROPOSED METHOD

In this paper, we propose a method for preserving the privacy of distribution information based on the method of [3]. As mentioned above, the proposed method assumes that parties can be uniquely identified by their blockchain addresses. Each party therefore encrypts the owner's blockchain address with the manufacturer's public key. This encrypted address is recorded in the blockchain to conceal the blockchain address and preserve privacy. The manufacturer can obtain blockchain addresses through decryption using its own private key and thereby track its products. Product owners and recipients share a secret token, using a zero-knowledge proof to demonstrate that they know it. The proposed method thereby guarantees distribution between correct owners and recipients while concealing blockchain addresses. The proposed method utilizes three smart contracts: a *Manufacturers Manager Contract* (*MMC*), which

manages manufacturer information; a *Products Manager Contract* (*PMC*), which manages product distribution; and a *Verifier Contract* (*VC*), which verifies the zero-knowledge proof. In what follows, after providing an overview of distribution using the proposed method, we present details in the following order: preparation for distribution, product enrollment, distribution management, and product tracking.

4.1 Overview

The following presents an overview of distribution using the proposed method in the case where product P is distributed by its manufacturer M, party X_1, and party X_2 in that order. As preparation for distribution, manufacturer M's blockchain address, public key, and company prefix are first registered in *MMC* (Section 4.2). To begin distribution, manufacturer M enrolls the EPC corresponding to product P in *PMC*. *PMC* obtains the company prefix by querying *MMC* with the blockchain address of the enroller. *PMC* accepts requests where the company prefix in the EPC matches the company prefix obtained from *MMC* (Section 4.3). Manufacturer M deploys *VC* on the blockchain for verification of the zero-knowledge proof in the receiving process described below. Manufacturer M runs a shipping process that designates the recipient of product P by recording the ciphertext of party X_1's blockchain address and the address of *VC* in *PMC*. At this point, *PMC* verifies that the party running the shipping process is manufacturer M. As a receiving process, party X_1 then sends a zero-knowledge proof to *PMC* to verify that X_1 is the designated recipient. *PMC* calls *VC* and receives a result. Only if the result is true, party X_1 is allowed to receive and the owner of product P recorded in *PMC* is updated with the ciphertext of party X_1's blockchain address. Party X_1 runs a shipping process that designates the recipient of product P by recording the ciphertext of party X_2's blockchain address and the address of *VC* in *PMC*. At this point, as in the previous receiving process, party X_1 sends a zero-knowledge proof to *PMC* to verify that X_1 is the current owner. Party then X_2 runs the receiving process, which is also the same as the previous receiving process by party X_1. As a result of that process, the owner of product P recorded in *PMC* is updated with the ciphertext of party X_2's blockchain address. Thereafter, product P is distributed through parties $X_3, X_4, \cdots X_n$ in the same way (Section 4.4). After distribution, the plaintext representing manufacturer M and the ciphertexts representing X_1 through X_n are recorded in *PMC*. Since the ciphertexts are generated with manufacturer M's public key, manufacturer M can obtain the plaintexts of blockchain addresses X_1 through X_n by decrypting them with its own private key. Manufacturer M is thus able to track product P (Section 4.5).

4.2 Preparation for distribution

As preparation for starting distribution, the manufacturer information is registered in *MMC*. This manufacturer information includes the manufacturer's blockchain address, public key, and company prefix. Other information such as the manufacturer's name and location can also be registered if necessary.

To prevent product distribution by unauthorized manufacturers, there should be a mechanism that allows only legitimate manufacturers to enroll their products on the blockchain. The manufacturer information is thus registered in *MMC*. During product enrollment (described below), only the manufacturer whose information is registered in *MMC* can run the enrollment process. An unauthorized party registering illegal information in *MMC* can begin to distribute products, so we assume that only a designated administrator can manage *MMC*. Candidate administrators are organizations having no conflicts of interest with other parties. One example is GS1 [17], a non-profit organization that designs and establishes international standards for supply chains such as EPC. Therefore,the proposed method assumes that *MMC* is centrally managed by GS1.

Note that *MMC* can be managed in a decentralized manner, thereby eliminating fraud by the administrator. However, there may be decision-making delays and difficulties in designing and operating this mechanism. In addition, since *MMC* information is recorded in the blockchain, anyone can monitor administrator behavior at will. In other words, anyone can check whether unauthorized manufacturers are registered, revealing malicious behavior by the administrator. Therefore, there is no need for decentralized management, and the proposed method assumes that *MMC* is centrally managed.

4.3 Product enrollment

The manufacturer runs the enrollment process by sending *PMC* a pair of digital signatures generated with its private key and the product's EPC. *PMC* obtains the manufacturer's blockchain address from the received digital signature and the corresponding company prefix from the blockchain address by querying *MMC*. The obtained company prefix is compared with the company prefix in the EPC. If it matches, the enrollment process is considered to have been run by the legitimate manufacturer, so the product is enrolled in *PMC*. If it does not match, the enrollment process is considered to have been run by an unauthorized manufacturer, so the enrollment process is rejected.

PMC uses the information in Table 3 to manage products. The information in Table 3 is managed as key–value pairs. Immediately after the product enrollment process, the manufacturer's blockchain address is set to the value of the *manufacturer* and *owner* keys. Originally, the encrypted blockchain address is recorded in the value field of key *owner*. Note, however, that since the manufacturer is not subject to privacy preservation, the blockchain address is recorded as plaintext. The product's EPC is also recorded in the value field of key *EPC*, because product information is managed with EPC. No other information in Table 3 is enrolled at this point. These key–value pairs are updated to appropriate values when the shipping and receiving process are run, as explained below. As a supplement, the proof used in the receiving process is recorded in value field of key *proof*. This value is used to confirm whether the proof for the receiving process is diverted during the shipping process.

4.4 Distribution management

Product distribution is managed by running the shipping and receiving processes for the products enrolled in *PMC*. Figure 4 illustrates the distribution management flow. Distribution management consists of nine steps, Steps 1 to 5 being related to the shipping process and Steps 6 to 9 being related to the receiving process.

1. The owner generates a secret token.

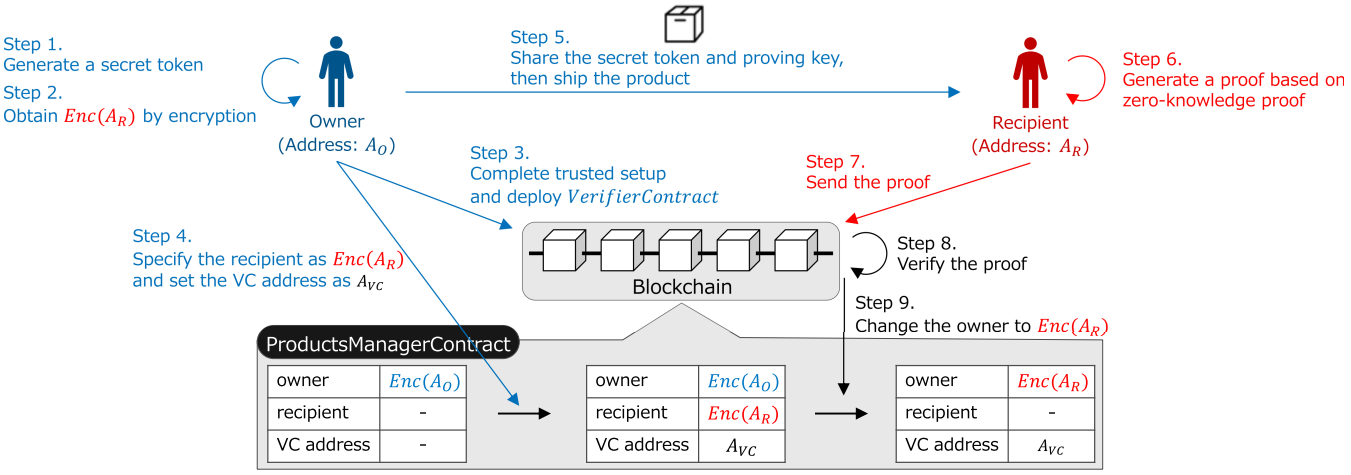

Figure 4: Distribution management flow in the proposed method.

Table 3: Product information recorded in *PMC*

Key	Value
EPC	EPC of attached product tag
Manufacturer	Manufacturer's blockchain address
Owner	Owner ciphertext
Recipient	Recipient ciphertext
VC address	*VC* contract address
Proof	Proof used in receiving process

2. The owner encrypts the recipient's address A_R with the secret token and the manufacturer's public key to obtain $Enc(A_R)$.
3. The owner generates a proving–verification key pair by Trusted Setup and deploys *VC* on the blockchain for proof verification.
4. The owner records the ciphertext of the recipient's address $Enc(A_R)$ and the contract address of *VC* in *PMC*.
5. The owner shares the secret token and proving key with the recipient by a secure method, then ships the product.
6. The recipient uses the shared proving key to demonstrate knowledge of the secret token shared in step 5 based on a zero-knowledge proof.
7. The recipient sends the proof to *PMC*.
8. *PMC* calls *VC* and verifies that the proof is valid.
9. The owner recorded in *PMC* is updated to $Enc(A_R)$.

PMC provides the functions necessary to perform Steps 4, 7, 8, and 9. The important steps are explained below.

First, we describe the address encryption in Step 2. We use the elliptic curve ElGamal encryption

$$Enc(A_R) = (kG, T + kQ) \tag{1}$$

to encrypt the address. Here, k is the secret token, G is the elliptic curve generator, T is the plaintext to be encrypted, and Q is the manufacturer's public key. The secret token is a 254-bit random number, which the owner generates by a cryptographically secure pseudorandom number generator at the time of each distribution.Only the owner and the recipient share this secret token. Note that all the operations in Equation (1) are on the elliptic curve. Therefore, plaintext T, which is the encryption target, must be transformed to a point T_p on the elliptic curve

$$T_p = (x_{T_p}, y_{T_p}), \tag{2}$$

where x_{T_p} and y_{T_p} are the values of x- and y-coordinates, respectively, on the elliptic curve. Following [18], to find x_{T_p} and y_{T_p} we generate 100 x_j values as

$$x_j = T \times 100 + j, \tag{3}$$

where $j \in \{0..99\}$. Then letting x_{T_p} be the smallest x_j that is a point on the elliptic curve, y_{T_p} can be computed from x_{T_p}.

$Enc(A_R)$ is recorded in the blockchain and can be browsed by anyone. Thus, if plaintext T is the recipient's address A_R, an attacker could successfully decipher it by investing sufficient computational resources. This does not preserve privacy. Therefore, the proposed method uses the exclusive-OR of the owner's address A_O and the recipient's address A_R. A successful attack therefore obtains $A_O \oplus A_R$, and identifying the recipient from this value requires knowledge of A_O. In other words, attackers wanting to identify the recipient at a given point in time must decipher all ciphertext used in the distribution up to that point, starting with the manufacturer. This is very difficult in terms of computational complexity, thus enhancing the strength of privacy preservation.

We next describe the zero-knowledge proof used in Step 3 and Step 6. While there are various implementations of zero-knowledge proofs, we utilize zk-SNARKs, which is known to be compatible with blockchain thanks to its non-interactivity and small proof size [19]. The following describes generation of the proving and verification key by Trusted Setup in Step 3 and the smart contract *VC*. Zk-SNARKs requires Trusted Setup, which generates a proving–verification key pair, to perform proofs. The proof generated with the proving key is verified with the verification key in that pair. In the proposed method, the

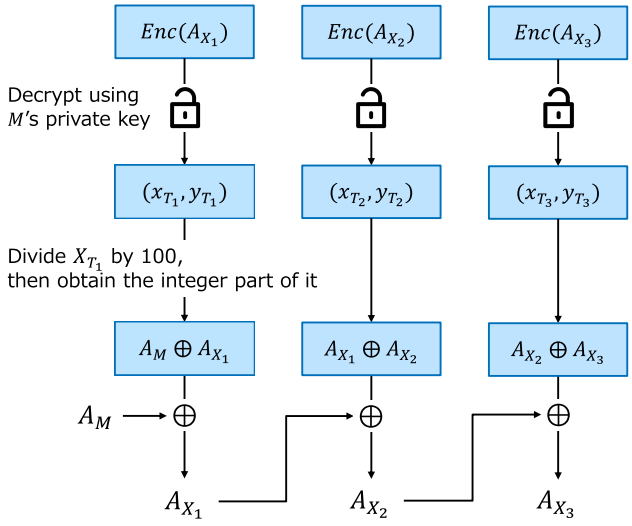

Figure 5: Only the manufacturer can obtain the ownership history.

owner performs Trusted Setup in Step 3, then deploys *VC* on the blockchain that verifies the proof with the verification key. The owner shares the proving key with the recipient in Step 5.

In Step 6, the recipient generates a proof demonstrating the computability of $Enc(A_R)$, based on the zero-knowledge proof with the proving key and secret token. Knowledge of the secret token is equivalent to being able to compute $Enc(A_R)$, as follows. Computing $Enc(A_R)$ requires knowledge of k, G, Q, T in Equation (1) and A_O, A_R for T. While G, Q, A_O and A_R are public information, only to the owner and recipient know k, or the secret token, and so are the only parties able to correctly compute $Enc(A_R)$. Recipients can thereby demonstrate their status as the designated recipient by computability of $Enc(A_R)$.

This proof is verified in a decentralized manner in Step 8 by *VC*. *PMC* calls *VC* with $Enc(A_R)$, the manufacturer's public key Q, and the proof generated in Step 6 as verification arguments. *VC* contains the verification key generated in Step 3. That is, only a proof correctly satisfying two conditions is accepted: it must have been generated with the proving key received in Step 5, and it must demonstrate the prover's ability to compute $Enc(A_R)$.

In practice, even in Step 4, the owner must prove ownership in the same manner. This prevents the shipping process from being run by someone other than the owner. Note that the owner generates this proof by reusing the values used at product receipt. That is, the ownership proof is based on the zero-knowledge proof demonstrating computability of $Enc(A_O)$ with the secret token and the proving key of a previous distribution part. The proof used when the owner received the product was stored in the value field of key *proof* in *PMC*. *PMC* uses this value to confirm that the same proof used in the receiving process is not diverted to this proof.

4.5 Product tracking

This section describes how manufacturers track their products. We assume that the product is distributed by manufacturer M, then parties X_1, X_2, and X_3 in that order. For convenience, suppose $M = X_0$. In this case, the product owner history recorded in *PMC* after the first shipment is $Enc(A_{X_1}), Enc(A_{X_2})$, and $Enc(A_{X_3})$, which are computed with the manufacturer's public key Q as

$$Enc(A_{X_i}) = (k_i G, T_i + k_i Q), \tag{4}$$

where $i \in \{1, 2, 3\}$, and k_i is the secret token. T_i is a point on the elliptic curve

$$T_i = (x_{T_i}, y_{T_i}), \tag{5}$$

where x_{T_i} and y_{T_i} are respectively x- and y-coordinate values on the elliptic curve.

Figure 5 shows how the manufacturer tracks the product distribution. Ciphertext $Enc(A_{X_1})$ was encrypted with the manufacturer's public key Q. Thus, the manufacturer M can obtain plaintext $T_1 = (x_{T_1}, y_{T_1})$ through decryption by its own private key. Manufacturer M obtains $A_M \oplus A_{X_1}$ by dividing x_{T_1} by 100, then obtains the integer part of the result. Manufacturer M knows its own address A_M. Therefore, manufacturer M can compute the exclusive-OR of $A_M \oplus A_{X_1}$ and A_M, thereby obtaining party X_1's address A_{X_1}.

Manufacturer M obtains plaintext $T_2 = (x_{T_2}, y_{T_2})$ from ciphertext $Enc(A_{X_2})$ in the same way as $Enc(A_{X_1})$, thereby obtaining $A_{X_1} \oplus A_{X_2}$ from x_{T_2}. Manufacturer M has obtained party X_1's address A_{X_1} in the previous step, and can therefore compute the exclusive-OR of $A_{X_1} \oplus A_{X_2}$ and the address A_{X_1} to obtain party X_2's address A_{X_2}. Manufacturer M similarly obtains party X_3's address A_{X_3} from ciphertext $Enc(A_{X_2})$. Thereafter, manufacturer M can similarly track product distribution through parties $X_4, X_5, \cdots, X_n$.

5. VERIFICATION

We verify traceability and privacy of the proposed method by considering fraudulent activities by attackers.

5.1 Traceability

There are four possible attack vectors for inhibiting traceability.

The first is to interfere with decryption of the owner's encrypted address using the manufacturer's private key. This can be performed by an attacker recording an encrypted statement on the blockchain using a public key other than the manufacturer's. However, the proof verification in Step 8 is performed by directly referring to the manufacturer's public key recorded in *MMC*, so the proof verification in Step 8 always fails if a statement encrypted with a public key other than the manufacturer's is recorded. Distribution by an attacker thus also fails, so this attack cannot succeed.

The second attack vector is for a third party not involved in distribution to carry out distribution by impersonating the owner or recipient. This could happen if the third party is able to generate a valid proof. However, those not knowing the information required for the proof cannot generate a valid proof because of the soundness of the zero-knowledge proof. Therefore, this attack cannot succeed.

The third attack vector is where the owner shares the secret token and proving key with an unauthorized recipient along with a legitimate recipient, which can result in unauthorized distribution. In this case, unauthorized recipients can correctly run the receiving process, because they know all information necessary to generate a valid proof. In this

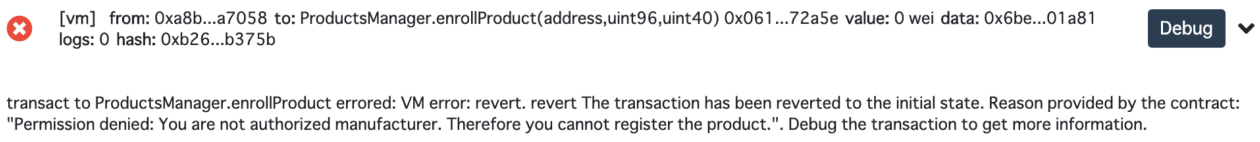

Figure 6: Results screen for Scenario 5.

context, there are three possible cases in terms of the product destination. The first case is where the product is sent to a legitimate recipient. In this case is the legitimate recipient fails to run the receiving process, because the unauthorized recipient has already done so. In this situation, it becomes immediately apparent that the owner has cheated when the legitimate recipient inquires with the manufacturer. There is thus no incentive for the owner to cheat. The second case is where the product is sent to a stranger. In this case, the stranger does not possess the information necessary to receive the product, so inappropriate deliveries are discovered and the product should be returned to the sender. The last case is where the product is sent to an unauthorized recipient. In this case, the unauthorized recipient can use the received information to run the receiving process. However, since that information is recorded in the blockchain, the manufacturer can later discover unauthorized distribution. There is no thus incentive for the owner to cheat. In each case, therefore, either there is no incentive to cheat or no significant problem arises. So while attacks are possible in theory, they are unlikely in practice.

The fourth attack vector is collusion between the owner and the recipient, making it difficult to identify the owner. This can be performed by an attacker using a made-up address or a real one belonging to someone else. The owner and recipient share an arbitrary address in addition to a secret token, and by using them to generate a proof, the owner information of PMC is updated correctly. It is therefore possible for this attack to succeed, and we intend to consider countermeasures against this attack in future work.

5.2 Privacy

An attacker can compromise privacy by retrieving addresses from the encrypted addresses or the proofs recorded in PMC.

We first consider the encrypted address. Cryptographic security depends on the key length and the cryptographic algorithm. For example, when using 254-bit elliptic curve cryptography, it is extremely difficult for a party who does not know the private key to decrypt the encrypted address in a practical time. We can thus ensure the security of the encrypted address in the proposed method by using the 254-bit elliptic curve proposed in [20].

We next consider the proof used in the proposed method. We use a zero-knowledge proof in the proposed method that is known to satisfy zero-knowledge-ness. In other words, it is not possible to recover information such as the address and the secret token from the proof. Attackers therefore cannot compromise privacy and thus cannot retrieve the owner's blockchain address.

6. EVALUATION

In this section, we implement the proposed method using Ethereum and confirm that the system behaves according to the model described in Section 3. System behaviors are examined under the scenarios described below. We also measure transaction fees and discuss a use case based on the results. This evaluation was conducted on September 9, 2020.

6.1 Environment

Our implementation used the following environment. We used the latest version of Ethereum at the time of this evaluation, as the public permissionless blockchain. We used version 0.6.2 of Solidity [21] to write smart contracts and the JavaScript Virtual Machine environment provided by Remix [22] to evaluate the proposed method. We used ZoKrates [23], a zk-SNARKs toolbox, for implementation of the zero-knowledge proof.

The system state before beginning the scenario was as follows. The blockchain address, public key, and company prefix of manufacturer M were registered in MMC. Manufacturer M held the EPC of its own product P. We assumed N parties, denoted as $X_i (i \in \{1..N\})$, were involved in the scenarios. For convenience, suppose $M = X_0$.

6.2 Scenarios

We run this implementation of the proposed method based on the following scenarios.

- Scenario 1: Normal product distribution
- Scenario 2: Confirming the manufacturer
- Scenario 3: Preserving privacy of distribution information
- Scenario 4: Distribution channel tracking by the manufacturer
- Scenario 5: Preventing distribution of counterfeit goods

In Scenario 1, we distribute a certain product three times, starting with manufacturer M and following parties X_1, X_2, and X_3 in order. In Scenario 2, we confirm that after execution of Scenario 1, any parties $X_j (j \in \{1..N\})$ can browse manufacturer M's address A_M recorded in PMC by product P's EPC. In Scenario 3, we confirm that after execution of Scenario 1, party $X_k (k \in \{1, 2, 3\})$ involved in the product distribution cannot know any distribution information other than manufacturer M, owner X_{k-1} who is their source of the product, and recipient X_{k+1} who is their destination of the product. We also confirm that party $X_l (l \in \{4..N\})$ not involved in the product distribution cannot know any distribution information other than manufacturer M. In Scenario 4, we confirm that after execution of Scenario 1, manufacturer M can identify all past to present owners

X_1, X_2 and X_3. In Scenario 5, we confirm that a party $X_m (m \in \{1..N\})$ who is not product P's manufacturer cannot begin its distribution. In this scenario, a party X_m tries to distribute product P as if it were the manufacturer.

6.3 Result

We first show the results of the scenarios. As a result of Scenario 1, manufacturer M successfully enrolled product P to *PMC*. Then, it was confirmed that product P's owner information transitioned as $Enc(A_{X_1}), Enc(A_{X_2}), Enc(A_{X_3})$ every time the shipping and receiving processes were run. As a result of Scenario 2, party X_j was able to browse A_M from *PMC* by using product P's EPC. In addition, party X_j could obtain detailed information about manufacturer M by querying *MMC*. As a result of Scenario 3, party X_k was able to obtain manufacturer M's information by using product P's EPC. In addition, party X_k could naturally obtain the information of parties X_{k-1} and X_{k+1}, because they were directly involved in each distribution. However, other distribution information stored in *PMC* was encrypted and could not be obtained, because there was no means of decryption. The same was true for X_l, where no information other than that for manufacturer M could be obtained. As a result of Scenario 4, manufacturer M was able to obtain the blockchain addresses of X_1, X_2, and X_3 in turn by the method described in Section 4.5. As a result of Scenario 5, enrollment of product P by X_m failed due to the error shown in Figure 6. These results indicate that the proposed method is able to preserve privacy of distribution information while preventing distribution of unauthorized products, and furthermore allowed product manufacturers to track their distribution.

Figure 7 shows transaction fees per party resulting from Scenario 1. In the proposed method, we found that transaction fees paid by other parties were higher than those of the manufacturer. In Ethereum, the cost of a function is generally determined by its instruction complexity and data size, with higher function costs resulting in higher transaction fees. First comparing the enrollment process and the shipping–receiving process for a product, the latter has more complex instructions, making its cost higher. We next compare destination and source information used in the shipment process. It is common for both manufacturers and other parties to use encrypted blockchain addresses as the destination information. However, the manufacturer can simply use its own blockchain address as source information, while the other parties need to use zero-knowledge proof information. The data size of zero-knowledge proof information is larger than that of the blockchain address, increasing the cost of its use. Parties other than the manufacturer therefore pay higher transaction fees. Specifically, total transaction fees paid by one party on the supply chain were at most 2.2×10^6 *gas* units.

6.4 Discussion

In the previous section we found that the total transaction fees required for one party was up to 2.2×10^6 *gas* units, which is not cheap. Converted to legal tender using the gas fee at the time of this evaluation, this is equivalent to 84.41 USD. Reducing fees by optimizing the implementation may be required for some cases, but even the current implementation can be applied to high-priced products such as automobiles and large home appliances. Such products

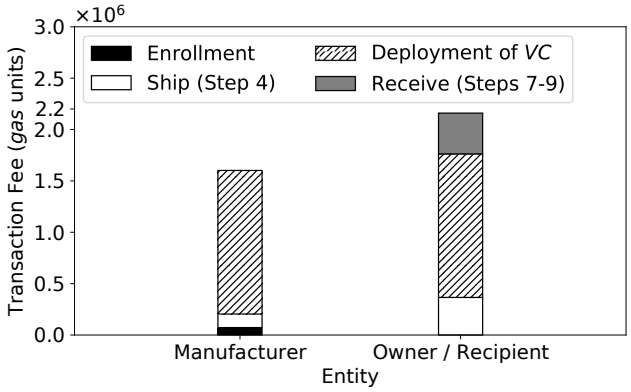

Figure 7: Per-party transaction fees for one product.

are subject to recall if the product has a problem or defect, creating challenges such as increasing consumer awareness of recalls and recall response rates [24]. In the case of distribution by the proposed method, only the manufacturer can track the product. Therefore, these issues can be solved by tracking any product that is subject to recall and recalling it through immediate notification to owners. In this case, the transaction fee may be regarded as a kind of warranty.

7. CONCLUSIONS AND FUTURE WORK

We proposed a method for using a public permissionless blockchain to track product distribution while preserving privacy in a supply chain. The proposed method preserves privacy by encrypting the blockchain address, which represents the owner information, with the public key of a manufacturer. The manufacturer of the product can thus track its distribution by decrypting the encrypted blockchain address with its own private key. To prevent unauthorized distribution, senders and recipients share a secret token, and the proposed method considers those knowing it as legitimate parties. Zero-knowledge proofs were used to demonstrate possession of the secret token without revealing owner information. We implemented the proposed method using Ethereum and verified that it works as expected. We also verified that the fee per person involved in the distribution was at most 2.2×10^6 *gas* units.

There are three outstanding topics for future works. The first is preserving privacy at the protocol level. In the proposed method, the blockchain address stored in the smart contract is encrypted and protected. Meanwhile, the blockchain protocol records the blockchain address of the smart contract's executor in blockchain. We assumed that the executor was the owner or recipient of the product, so anyone can identify the owner from the smart contract's execution history; that is, privacy is not preserved at the protocol level. One possible solution is to introduce an intermediate server that executes smart contracts on behalf of the owner or recipient to avoid direct smart contract execution by them.

The second topic requiring further investigation is increasing the scope of the proposed method by reducing transaction fees to allow its application to less expensive products. Figure 7 shows that the deployment process of *VC* accounts for most of the transaction fees. To that end, we will consider a strategy in which the product manufacturer deploys *VC* once and other parties repeatedly distributing the same

product reuse that *VC*, instead of deploying it for each distribution. This will significantly reduce transaction fees, because *VC* will be deployed only once per product.

The third topic is extending the method so that it can deal with product assembly and disassembly. Large products are often composed of several smaller finished products or modules. Extending the proposed method will allow identifying large products with defective modules and verifying the authenticity of reused modules.

8. REFERENCES

[1] OECD/EUIPO, *Trade in Counterfeit and Pirated Goods*. Illicit Trade, Paris: European Union Intellectual Property Office, Mar. 2019.

[2] Centers for Disease Control and Prevention, "Multistate outbreaks of Shiga toxin-producing Escherichia coli O26 infections linked to Chipotle Mexican Grill Restaurants (final update)." https://www.cdc.gov/ecoli/2015/o26-11-15/index.html, Feb. 2016. [Online; accessed 18-September-2020].

[3] K. Toyoda, P. Takis 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, June 2017.

[4] H. M. Kim and M. Laskowski, "Toward an ontology-driven blockchain design for supply-chain provenance," *Intelligent Systems in Accounting, Finance and Management*, vol. 25, pp. 18–27, Mar. 2018.

[5] H. Huang, X. Zhou, and J. Liu, "Food supply chain traceability scheme based on blockchain and EPC technology," in *Proceedings of Smart Blockchain*, pp. 32–42, Nov. 2019.

[6] V. Acharya, A. E. Yerrapati, and N. Prakash, *Oracle Blockchain Quick Start Guide: A practical approach to implementing blockchain in your enterprise*. Packt Publishing Ltd, Sept. 2019.

[7] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with IoT. challenges and opportunities," *Future Generation Computer Systems*, vol. 88, pp. 173–190, Nov. 2018.

[8] L. Insights, "Coca Cola bottlers to trial public Ethereum for supply chain transparency." https://ledgerinsights.com/coca-cola-bottlers-coke-blockchain-ethereum-baseline/, Aug. 2020. [Online; accessed 18-September-2020].

[9] N. Hackius and M. Petersen, "Blockchain in logistics and supply chain: Trick or treat?," in *Proceedings of the Hamburg International Conference of Logistics (HICL)*, pp. 3–18, Oct. 2017.

[10] N. Kshetri, "1 Blockchain's roles in meeting key supply chain management objectives," *International Journal of Information Management*, vol. 39, pp. 80–89, Apr. 2018.

[11] F. Tian, "An agri-food supply chain traceability system for China based on RFID blockchain technology," in *Proceedings of 2016 13th International Conference on Service Systems and Service Management (ICSSSM)*, pp. 1–6, June 2016.

[12] D. Sinclair, H. Shahriar, and C. Zhang, "Security Requirement Prototyping with Hyperledger Composer for Drug Supply Chain: A Blockchain Application," in *Proceedings of the 3rd International Conference on Cryptography, Security and Privacy*, pp. 158–163, Jan. 2019.

[13] "IBM Food Trust." https://www.ibm.com/blockchain/solutions/food-trust. [Online; accessed 18-September-2020].

[14] M. el Maouchi, O. Ersoy, and Z. Erkin, "DECOUPLES: A Decentralized, Unlinkable and Privacy-Preserving Traceability System for the Supply Chain," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pp. 364–373, Apr. 2019.

[15] "Baseline Protocol." https://docs.baseline-protocol.org/. [Online; accessed 18-September-2020].

[16] V. Buterin, "Ethereum Whitepaper." https://ethereum.org/en/whitepaper/, 2013. [Online; accessed 18-September-2020].

[17] "GS1." https://www.gs1.org/. [Online; accessed 18-September-2020].

[18] L. C. Washington, *Elliptic Curves: Number Theory and Cryptography, Second Edition*. Chapman and Hall/CRC, Apr. 2008.

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

[20] B. WhiteHat, J. Baylina, and M. Bellés, "Baby Jubjub Elliptic Curve." https://iden3-docs.readthedocs.io/en/latest/_downloads/33717d75ab84e11313cc0d8a090b636f/Baby-Jubjub.pdf. [Online; accessed 18-September-2020].

[21] "Solidity." https://solidity.readthedocs.io/. [Online; accessed 10-February-2020].

[22] "Remix - Ethereum IDE." https://remix.ethereum.org. [Online; accessed 18-September-2020].

[23] "ZoKrates." https://github.com/Zokrates/ZoKrates. [Online; accessed 18-September-2020].

[24] OECD, "Enhancing product recall effectiveness globally," *OECD Science, Technology and Industry Policy Papers*, Nov. 2018.