Internet of Things 11 (2020) 100253

Contents lists available at ScienceDirect

Internet of Things

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

Research article

Scalable and secure product serialization for multi-party perishable good supply chains using blockchain

Subhasis Thakur*, John G. Breslin

National University of Ireland, Galway, Ireland

ARTICLE INFO

Article history:
Received 10 May 2020
Revised 23 June 2020
Accepted 24 June 2020
Available online 26 June 2020

2010 MSC:
00-01
99-00

Keywords:
Serialization
Blockchain
Scalability
Traceability

ABSTRACT

Product serialization aims to allocate unique serial numbers to products in a supply chain. The security challenges to product serialization are: • Valid serial numbers can be stolen and used to label fake products. Thus uniqueness of a serial number should be verifiable at any stage of its lifecycle in a supply chain. • A planned change of custody of a product in distribution can be corrupted by a few intimidatory nodes. Compliance with the planned change of custody should be verifiable. • The manufacturer and the consumer should be able to verify that perishable food products with expired shelf life are discarded.

In this paper, we use blockchains to develop a product serialization method that solves the above security issues in a multi-party perishable good supply chain. Blockchains can revolutionize security and transparency in supply chains by providing a secure data-sharing platform in a multi-party environment. Although blockchains can provide a secure data storage of change of custody events of products in a supply chain, a high volume of such events poses scalability problems for blockchains. In this paper, we solve the product serialization problem using blockchain offline channels. Our solution significantly reduces the number of transactions needed to be recorded in the blockchain. We propose a secure serialization protocol to verify the authenticity of serial numbers despite not frequently engaging with the blockchain.

1. Introduction

Product serialization intends to allocate unique serial numbers to products in a supply chain. Unique serial numbers can be generated by the manufacturer or by a regulator who can assign serial numbers to a manufacturer and the manufacturer can use these numbers to serialize its product. Serialization is important to ensure product authenticity and safety. Although interoperable data format for serialization and data standard [1] are in use for product serialization, the current state of the art in product serialization lacks security evident from a high volume of counterfeit products. It is reported [2] that annually global businesses lose up to US$597 billion per year due to fake products and counterfeiting will lead to the loss of 5.4 million jobs globally. Counterfeit medicines [3] costs US$200 billion annually and up to 10 million people die from fake medicine every year. The security challenges [4] to product serialization are the following:

* Corresponding author.
E-mail addresses: subhasis.thakur@nuigalway.ie (S. Thakur), john.breslin@nuigalway.ie (J.G. Breslin).

https://doi.org/10.1016/j.iot.2020.100253

- **Security of serial numbers:** Valid serial numbers can be stolen and used to label fake products. Thus uniqueness of a serial number should be verifiable at any stage of its lifecycle in a supply chain.
- **Secure Change of Custody:** A planned change of custody of a product in distribution can be corrupted by a few intimidatory nodes. Compliance with the planned change of custody should be verifiable by the consumer of the product.
- **Control over the serial number - Perishable Food:** The manufacturer should be able to ensure that perishable food products with expired shelf life are discarded.
- **Multi-party supply chain:** In this paper, we consider a multi-party supply chain where no party has absolute control over the entire supply chain. We consider regulators, manufacturers, intermediators (logistics service providers, cold storage providers etc), retailers, consumers as members of the supply chain. In such a federated supply chain, parties may not trust each other and have various levels of concern for the first three challenges. For example, regulators may be only concerned about the security of serial numbers to prevent counterfeit products. The manufacturer may be primarily concerned about discarding perishable goods with expired shelf life. The intermediator may be concerned about the proper change of custody of the product.

In this paper, we use blockchains to develop a product serialization method that solves the above security issues. Blockchains can revolutionise security and transparency in supply chains by providing a secure data-sharing platform in a multi-party environment. Although blockchains can provide a secure data storage of these events, a high volume of such events poses scalability problems for blockchains. In this paper, we solve the product serialization problem using offline channels of blockchains. Our solution significantly reduces the number of transactions needed to be recorded in the blockchain. We propose a secure serialization protocol to verify the authenticity of serial numbers despite not engaging with the blockchain.

The state of art blockchain-based traceability solutions for perishable good supply chains uses IoTs to monitor the supply chain and smart contracts to ensure traceability in the supply chain. However, the state of art solutions do not solve the above-mentioned problems. In this paper, we mitigate these problems as our main contributions are as follows:

Secure serialisation: We have developed a product serialization method which prevents a counterfeiter to copy a genuine serial number to label a counterfeit product. The existing traceability solutions for food supply chains may analyze the transactions to detect non-uniqueness of serial numbers. Thus these solutions do not prevent copying genuine serial numbers.

Trust in multi-party supply chains: In a multi-party supply chain, various parties have trust issues as a party wants to ensure that another party has shipped genuine products and another party has not shipped products with expired shelf life. The existing blockchain-based supply chain management solutions do not solve these trust issues.

Control for perishable goods: We present a serialization solution that prevents the distribution of products with expired shelf life. The existing blockchain-based supply chain management solutions do not prevent the distribution of products with expired shelf life.

Secure change of custody: Our serialization solution ensures that a pre-defined distribution path is followed for a product. The existing blockchain-based traceability solutions can prove whether or not a pre-defined distribution path is followed. But it can not enforce the movement of a product in a supply chain in a predefined path.

Scalability: Our solution is scalable as it uses offline channels in a blockchain. Existing solutions except Hyperledger-based solutions are not scalable. Moreover, Hyperledger uses proof of authority which requires a certain level of trust among the parties as only a fraction of peers of the blockchain act as transaction validators.

The paper is organised as follows: in Section 2 we describe the blockchain and offline channels used in this paper, in Section 3 we define the serialization security issues as trust problems among various parties in a supply chain, in Section 4 we describe our serialisation method with blockchains, in Section 5 we analyse security of the proposed solution, in Section 6 we discuss related literature, and we conclude the paper in Section 7.

2. Problem statement: serialization in supply chains

In this paper, we will consider a multi-party supply chain [5] composed of a regulator, a manufacturer, a set of intermidiator parties (logistics, cold-storage providers, etc.), a retailer and a set of customer. We consider the scenario when none of these parties has absolute control over the supply chain. The regulator is responsible for assigning valid serial numbers to the manufacturer, the manufacturer is responsible for labelings of the products with these serial numbers, the intermediator nodes are responsible for moving the products through the distribution line, the retailer is responsible for selling the product before its expiry date (Fig. 1).

We will represent a supply chain network as a directed graph $G = (V, E)$ where nodes indicate parties (distributors, suppliers, wholesalers, retailers) engaged in a supply chain and edges indicate the possible path of change of custody of a product. $P = (p_1, \ldots, p_n)$ be a set of products to be distributed in the supply chain G. A distribution plan of a product p_i is a path in G which indicates the sequence of change of custody of p_i. The distribution plan of a product is known by at least one party in V. $\delta(p_i) \subset (v_i \times v_j)$ will denote the distribution plan for the product p_i. $\theta(p_i)$ will denote the planner of the product p_i. $\theta(p_i)$ will be responsible to ensure that p_i moves according to the plan $\delta(p_i)$. Each path $\delta(p_i)$ is a directed acyclic weakly connected subgraph of G. $Id = (i_1, \ldots, i_m)$ be the set of serial numbers to be assigned with the products.

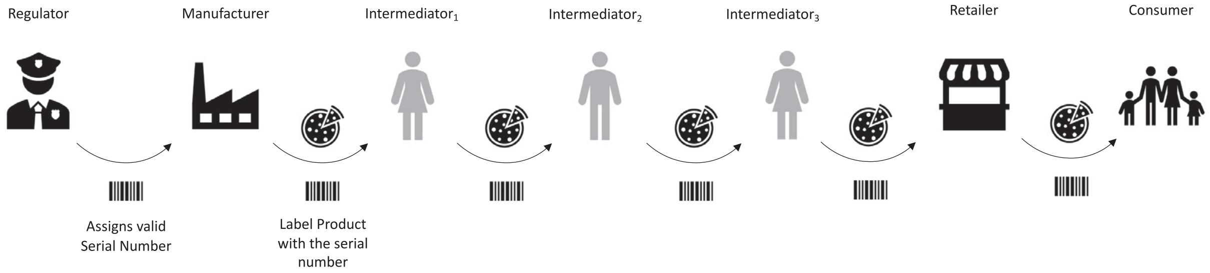

Fig. 1. A multi-party supply chain. The regulator is responsible for physical movement serial numbers to the manufacturer, the manufacturer is responsible for the movement of labeled products to the intermediators, the intermediators are responsible for the movement of labelled products to the retailer and the retailer is responsible for the movement of labelled products to the consumer.

Definition 1. A supply chain will be represent as the tuple $SC =< G = (V, E), P, ID, \theta, \delta >$.

Definition 2. In a supply chain $SC =< G, P, ID, \theta, \delta >$, products are following proper change of custody if any party v_i in a distribution plan $v_1 \rightarrow v_2 \rightarrow \dots v_n$ can verify that the actual product movement followed the path $v_1 \rightarrow v_2 \rightarrow \dots v_{i-1}$ as per the distribution plan.

Definition 3. Serialisation procedure in a supply chain $SC =< G, P, ID, \theta, \delta >$ is consists of methods to assign and retrieve serial numbers of the products in a supply chain. A serialisation procedure is correct if all products are following proper change of custody.

In this paper, we develop a correct serialisation method for a multi-party supply chain. In a multi-party supply chain, there are trust issues among the parties and a correct serialisation method must address these trust issues.

2.1. Trust problem 1

The regulator does have control over the supply chain. Its objective is to assign a unique serial number of each product. Further, in the case of perishable food supply chains, it wants to ensure that food products are sold before their expiry dates. Its objective is based on reducing counterfeit items from the market and reduce the social implications of counterfeit perishable food products such as death from fake medicine or disease outbreak from consuming counterfeit foods.

Adversary: The models of adversaries are:

- *Adversarial Manufacturer:* the manufacturer may label multiple products with the same serial number.
- *Adversarial Intermediator:* The intermediator parties may steal the genuine serial number from a product in the distribution line and use it to label and inject fake products into the supply chain.
- *Adversarial Retailer:* The retailer may collude with an adversarial manufacturer to label counterfeit products with stolen serial numbers.

Mitigation: The regulator can trust the supply chain if it is not possible to use a serial number more than once to label products and in case of perishable food products, the products must be sold before its expiry date.

2.2. Trust problem 2

The concern of the manufacturer in a multi-party supply chain is the proper change of custody of the product, sale of the product before its shelf life expires, and prevention of fake products labelled with stolen products.

Adversary: The models of adversaries are:

- *Adversarial Intermediator:* The intermediator parties may steal the genuine serial number from a product in the distribution line and use it to label and inject fake products into the supply chain. Such actions will impact the revenue of the manufacturer. Also, the intermediators may not follow the planned change of custody of a product. Such disruption in the change of custody may impact the quality of the product. For example, the manufacturer may plan that a product should stay in cold storage as it is transferred from location to another location. But the intermediator nodes may skip the cold storage and this may lead to decay in product quality.
- *Adversarial Retailer:* The retailer may sell the product after its expiry date. It may cause health issues of the consumers and such events will impact the brand value of the manufacturer.

Mitigation: The manufacturer can trust the supply chain if (a) it is not possible to use a serial number more than once to label products and in case of perishable food products, (b) it is not possible to disrupt planned change of custody and (c) not possible to sell the products after its expiry date.

2.3. Trust problem 3

The intermediators want to distribute products with genuine serial numbers, which are not expired.

Adversary: The models of adversaries are:

- *Adversarial manufacturer / Intermediator:* An intermediator must ensure that it has received a product according to a correct planned change of custody.
- *Adversarial Retailer:* The intermediator must ensure that it has received products before its shelf life has expired.

Mitigation: The intermediators can trust the supply chain if (a) it is not possible to disrupt the planned change of custody and (b) it is not possible to receive products with expired shelf life.

2.4. Trust problem 4

The retailer wants to sell genuine and not-expired products.

Adversary: The models of adversaries are:

- *Adversarial manufacturer / intermediator:* The manufacturer or the intermediator may distribute fake products labelled with stolen serial numbers.
- *Shelf life of products:* The intermediator may give the retailer products with expired shelf life.

Mitigation: The retailer can trust the supply chain if (a) it is not possible to disrupt the planned change of custody and (b) it is not possible to receive products with expired shelf life.

2.5. Trust problem 5

The consumer wants to buy genuine and not expired products.

Adversary: The models of adversaries are:

- *Adversarial Retailer:* The retailer may sell multiple products with the same serial number or may sell products after their shelf life expires.

Mitigation: The retailer can trust the supply chain if it is not possible to use a serial number more than once to label products and in case of perishable food products, the products have not reached their expiry dates before they deliver it to the retailer. The consumer should be able to check if the supply chain has exhibited a proper change of custody to move its product.

2.6. Scalability problem

In this paper, we use blockchains to develop a correct serialisation method in a multi-party supply chain. Proof of work or Proof of stake based blockchains have scalability problems. In a large supply chain with numerous products in its distribution line scalability is a significant issue. We use offline channels to mitigate scalability problems. In the next section we describe offline channels used in Bitcoin's Lightning network. In this paper, we will use similar offline channels for serialization.

3. Blockchains and offline channels

3.1. Blockchains

Simplified workflow of proof of work or proof of stake-based blockchains is as follows (Fig. 2):

1. Transactions are created by peers usually using unspent-transaction-output rules which states that to create a new transaction it must have input transactions which are not used in any other transaction as the input transactions and value of the new transaction is at most the total value of the input transactions. Although there are other blockchain systems which use different transaction creation rules.
2. After creating the transaction the peer publishes the transaction to the blockchain network. This new transaction will be added in a block by a miner.
3. After creating a new block a miner solves the proof of work puzzle by producing a random string such that its Hash has a predefined pattern. After solving the mining puzzle, the miner publishes the new block to the blockchain network.
4. Other miners, upon receiving the new block will verify the transactions in this block and verify if the puzzle has solved correctly. If the new block is correct then it will be added to the blockchain.
5. Each block contains a parent block information. After verifying the new block, a miner will add the new block as the child block of the block labelled as the parent block in the new block.
6. The new block will be regarded as the blockchain head if its distance from the first block is more than any other block.
7. It may be possible that the new block's parent block is not the last known blockchain head. In such a case, if the distance from the new block to the first block is more than any other block then the miner will regard the shortest path from the new block to the first block as the valid blockchain.

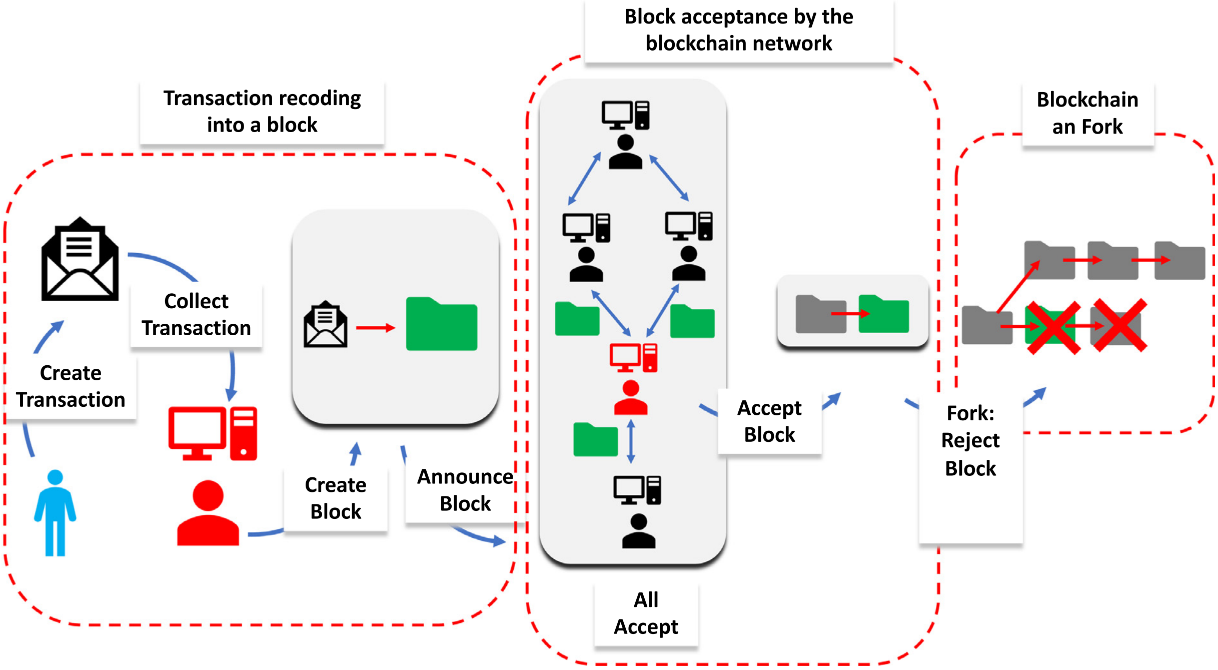

Fig. 2. Sequence of events in a blockchain network.

3.2. *Offline channels*

The basic protocol for using an offline channel (for Bitcoin Lightning network [6]) is as follows (shown in (Figs. 3 and 4)):

1. Offline channels use Hashed Time Locked Contracts[1] to create and update channels.
2. Say Alice and Bob wants to create a channel between them with balances 10 tokens (each contributes 5 tokens).
3. Alice and Bob creates two pairs of lock (hash) and key (random string). They exchange the locks.
4. Bob creates a 'confirmation transaction' as follows:
   (a) There is a multi-signature address between them which requires a signature from both to transfer funds from it. We will call this address M_1.
   (b) Bob creates transactions from M_1 which states that Bob will get 5 tokens and the remaining 5 tokens will go to another multisignature address between them. We will call this address M_2.
   (c) The 5 tokens in M_2 will be given to Alice after 10 days or Bob can claim it if it can produce the key to the lock of Alice.
5. Bob signs this transaction and sends it to Alice who can use it to get tokens from the channel by signing it and publishing it to the blockchain network.
6. Alice produces a mirrored confirmation transaction and sends it to Bob. The confirmation transactions ensure that both parties can recover from if they fund the channel between them.
7. Now Alice and Bob transfer funds in the multi-signature address by creating transactions in the blockchain and hence the channel becomes operational.
8. Both parties should exchange keys and create new confirmation transaction to update the channel. They do not need to update the blockchain as the update confirmation transactions.
9. If any party announces a confirmation transaction then the channel closes.

Further, the offline channel network supports Path-Based Fund Transfer (PBT). A PBT uses a path between two parties in a channel network for funds transfer between them. The path is a collection of channels and PBT allows peers without a mutual channel to transfer fund in offline. A PBT protocol [6] for this offline channel network is as follows:

1. Say Alice wants to send funds to Carol via Bob.
2. Carol will create a lock and a key.
3. In the multi-signature address between Carol and Bob, a contract will be created as follows:
   (a) Bob will send 5 tokens to this address.

[1] https://en.bitcoin.it/wiki/Hash_Time_Locked_Contracts

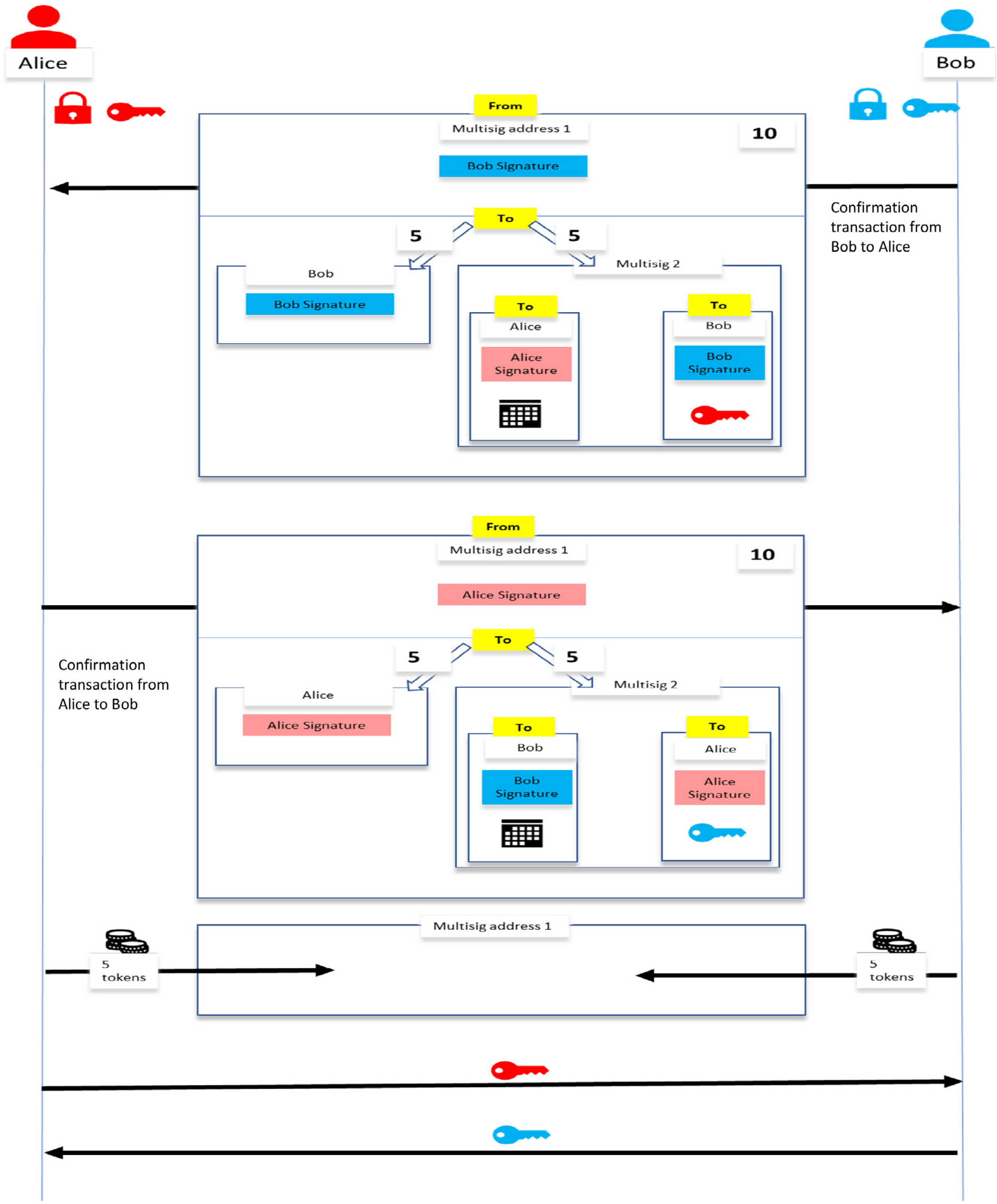

Fig. 3. Protocol for using offline channels.

   (b) Bob will get these tokens back after 9 days if Carol does not claim it.
   (c) Carol can claim it anytime if it can produce the key to the lock.
4. Similarly, another contract will be created between Alice and Bob as follows:
   (a) Alice will send 5 tokens to this address.
   (b) Alice will get these tokens back after 10 days if Bob does not claim it.
   (c) Bob can claim it anytime if it can produce the key to the lock.
5. Thus Carol reveals the key to Bob as it collects the fund, which Bob uses to get refunded from Alice.

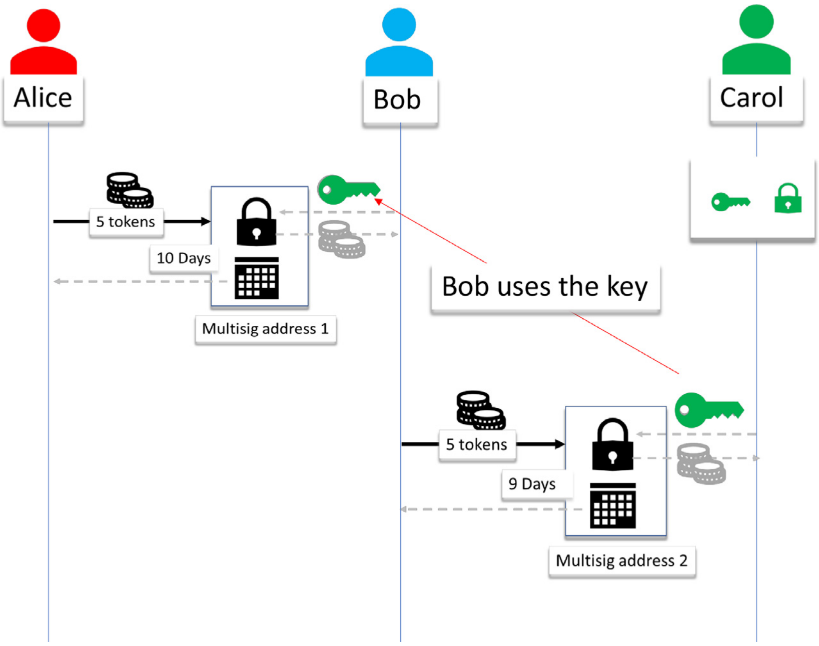

Fig. 4. Protocol for path based transfer.

4. Secure serialisation with offline channels

In this paper, we will use proof-of-work based consensus model for blockchains and Bitcoin Lightning network as a model of offline channels. The blockchain network is consists of nodes from the regulator, the manufacturer, the intermediators, the retailers, and the consumers. Briefly, our serialisation solution is as follows (Fig. 5):

- We introduce a concept of local serial numbers between pairs of parties who are involved in an immediate change of custody of products. The local serial numbers are used to record the change of custody event between two parties. We use bi-directional offline channels similar to Bitcoin Lightning network as discussed in Section 3, to generate these local serial numbers. The main blockchain records the creation of channels, i.e., the initial assignment of serial numbers but does not need to record the updated assignment of local serial numbers. For example, an offline channel between parties A and B records that, A has the local serial numbers x_1, x_2. An updated channel may indicate that A has serial number x_1 and B has serial number x_2, i.e., A has transferred serial number x_1 to B. It means there was a product with the label x_1 and A sent it to B.
- Next, similar to Path-based fund transfer, we will use a path among all parties in a planned change of custody for such local serial number transfer among pairwise parties.
- First, the regulator sends a genuine serial number to the manufacturer using such an offline channel and the manufacturer labels a product with the serial given by the regulator.
- Next, the manufacturer receives a request for a product from the retailer and the payment for it. The manufacturer forms a planned change of custody of the product and informs the retailer about the identification number of the product to be sent and planned change of custody.
- We develop a protocol to record the change of custody from the manufacturer to the retailer via intermediator nodes using offline channels and local serial numbers among them. The protocol is similar to PBT.
- Next, once a consumer buys a product, the PBT in the previous step is extended as the consumer becomes the last node (recipient of a PBT). The consumer initiates the execution of the PBT. If all contracts are executed from the consumer to the regulator then it means that the product has followed the proper change of custody.
- We use Hashed Time Lock Contracts and time in these contracts is at most the shelf-life of a product. Hence a successful PBT execution will happen before a product's shelf life expires.
- The regulator issues a set of unique serial numbers. The manufacturer creates PBTs only using unique serial numbers hence, serial numbers can not be reused i.e., stolen and used in counterfeiting.

Next, we present a detailed description of the above-mentioned procedure: First we will discuss how to create channels for local serial numbers. Next, we will show how a regulator sends the serial number to the manufacturer, how the manufacturer records change of custody information to the retailer, and how the consumer is included in change of custody events and how it verifies a proper change of custody by executing a PBT.

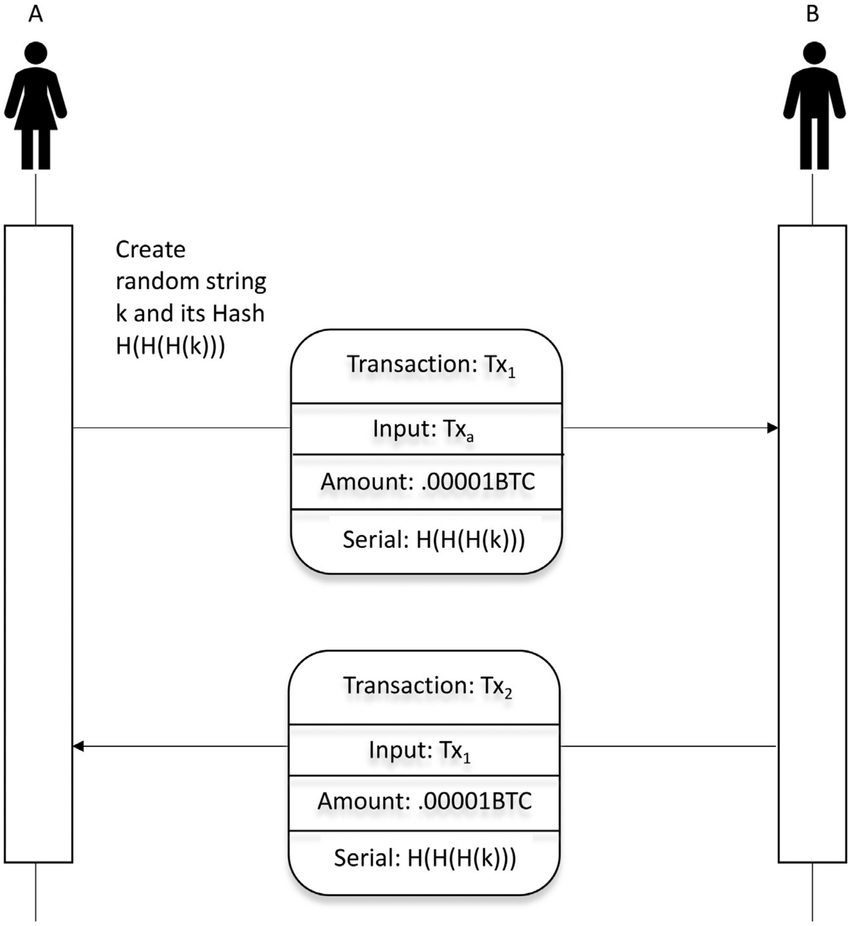

Fig. 5. *Protocol*$_1$ Protocol for local serial numbers.

4.1. Creation of local serial numbers

In this paper, we will use proof of work-based blockchains such as Bitcoin. However, the proposed solution can be implemented in any model of blockchains that allows the creation of a multi-signature address. This is because we will need multi-signature addresses to create an offline channel.

We will use the Bitcoin transaction data structure with additional field serial numbers which can be empty. This additional data field does not change the transaction acceptance and consensus principles of the Bitcoin blockchain model. $Protocol_1$ assigns secret local serial numbers to the parties in a supply chain. A party A may be assigned a set of local serial numbers which satisfies the following properties:

- These serial numbers are only known to A. The blockchain network records A's commitment to these numbers, i.e., A can choose the local serial numbers and does not reveal it to anyone but it can not modify these numbers.
- If there is a change of custody of a product from A to B using a local serial number $H(H(H(k)))$ then B can only prove that it has received the product from A if B can present k.
- In $Protocol_1$ A creates a serial number by creating a random text k and it's Hash $H(k)$. A uses $H(k)$ in a transaction tx_1 to B. B sends back these serial numbers to A by creating a transaction tx_2 whose input is tx_1. B can be any node in the blockchain or A can create a new transaction to itself to generate new serial numbers.
- A party can only use local serial numbers if it is mentioned in unspent transactions and only this party knows the key to this serial number.
- Further, it is required that Tx_1 is unspent and all parties in a supply chain will be given a fixed number of unspent transactions (with random serial numbers) which they will use to generate the first set of serial numbers.

4.2. Creation of offline channels for serial numbers

The protocol for creating a new channel between two parties A and B is as follows (shown in Fig. 6):

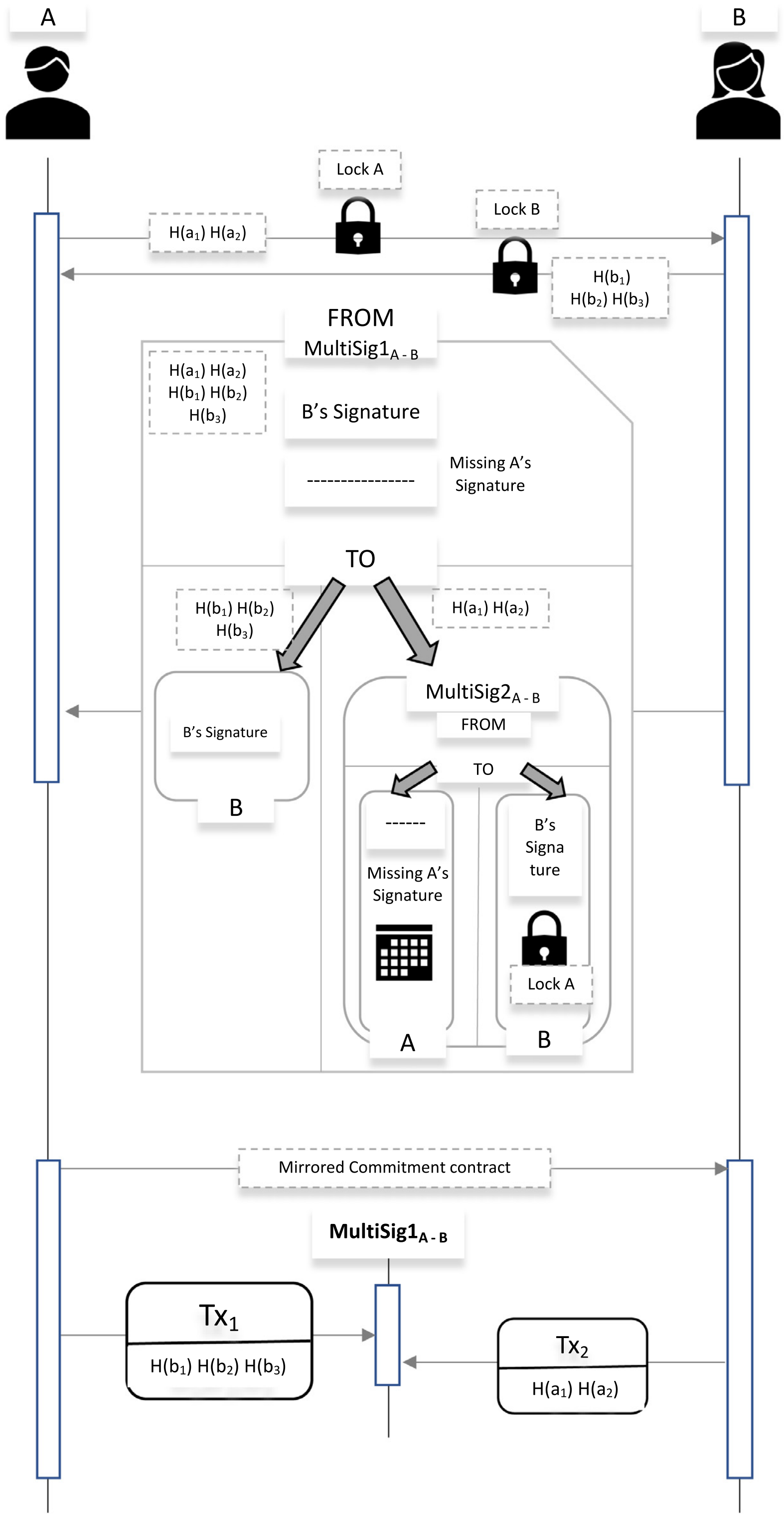

Fig. 6. Protocol for opening channel to exchange local serial numbers.

1. First, both parties inform each other about the local serial numbers to be used in their channel. In this case, A informs B about $H(a_1)$, $H(a_2)$ and B informs A about $H(b_1)$, $H(b_2)$, $H(b_3)$. Note that the actual local serial number is a_1 and A informs B about Hash of a_1.
2. A creates a key K_A (random string) and Lock $H(K_A)$ (Hash of K_A). A informs B about $H(K_A)$. Similarly, B informs A about $H(K_B)$.
3. B prepares a Hashed Time Lock Contract as follows:
   (a) From a multi-signature address $MultiSig1_{A-B}$ between A and B (it requires the signature of both parties to transfer from this address), B will get the serial numbers $H(b_1)$, $H(b_2)$, $H(b_3)$ immediately if the contract is published to the blockchain. The serial numbers $H(a_1)$, $H(a_2)$ will be given to another multi-signature address $MultiSig2_{A-B}$ between A and B.
   (b) From the address $MultiSig2_{A-B}$ serial numbers $H(a_1)$, $H(a_2)$ will be given to A after time x (may be calculated as the number of new blocks created) or to B if B can produce K_A.
   (c) B sends this contract (called the confirmation transaction) to A.
4. A creates a mirrored confirmation contract and sends it to B.
5. After receiving confirmation contracts the parties fund $MultiSig1_{A-B}$ with transactions Tx_1 (from A and with local serial numbers $H(b_1)$, $H(b_2)$, $H(b_3)$) and Tx_2 (from A and with local serial numbers $H(a_1)$, $H(a_2)$). As these transactions are recorded in the blockchain, the channel between A and B becomes operational.

Next, the protocol to update a channel is as follows (shown in Fig. 7):

1. First, both parties exchange the key for their respective locks in the last confirmation transaction. Next, they prepare updated confirmation transactions.
2. Both parties create a new lock and key and share the locks.
3. Say B wants to send serial number b_1 to A. It creates a new confirmation transaction as follows:
   (a) From the address $MultiSIg1_{A-B}$, B immediately gets $H(b_2)$, $H(b_3)$ if the transaction is published in the blockchain network.
   (b) $H(b_1)$, $H(a_1)$, $H(a_2)$, go to $MultiSig2_{A-B}$ such that: A will receive all serial numbers from $MultiSig2_{A-B}$ after x time (which is less than the time used in last confirmation transaction between them). B can all serial numbers from $MultiSig2_{A-B}$ if it can produce the lock of A and it can produce b_* such that $H(b_*) = H(b_1)$. B sends this confirmation transaction A.
4. A creates a mirrored confirmation transaction and sends it to B.
5. For the next update of the channel, B reveals b_1 to A. This prevents B from reclaiming $H(b_1)$ by publishing old confirmation transactions.

4.3. Regulator to manufacturer

The serialisation protocol for transferring serial numbers from the regulator to the manufacturer is as follows (Illustrated in Fig. 8):

- We will use channel creation and updating protocol for local serial number transfer for secure transfer of serial number from the regulator to the manufacturer.
- The initial channel balance between them is as follows: Let $H(k_1), \ldots, H(k_n)$ are the Hash of serial numbers $k_1, \ldots, k_n$. According to the first HLTC of this channel, all serial numbers will be assigned to the regulator's account after a finite timeout.
- The manufacturer can request a set of serial numbers from the regulators and upon receiving such a request, the regulator will update the channel. For example, by updating the channel as serial numbers $H(k_2), \ldots, H(k_n)$ belong to the regulator and the serial number $H(k_1)$ belongs to the manufacturer, the manufacturer reveals k_1 to the manufacturer.
- Once all IDs are transferred, the last confirmation transaction is published to the blockchain network. Hence the blockchain records change of custody of valid serial numbers to the manufacturer.
- After publishing the last confirmation transaction to the blockchain network, i.e., closing the channel, both parties again open another channel with a new set of serial numbers.

4.4. Manufacturer to retailer

The protocol for serialisation for transferring serial numbers from the manufacturer to the retailer is as follows:

- The retailer places the order of a product by paying for it to the manufacturer. After receiving the payment, the manufacturer prepares a planned change of custody of the product and informs the retailer about $H(K_1)$ (K_1 is the serial number that the manufacturer received from the regulator) and sequence of public keys of all parties in the planned change of custody of the product.
- The manufacturer communicates and creates a sequence of HTLCs where each intermediator participates in two HTLCs. In each of these HTLC there are two locks $H(H(K_1))$ and $H(E(P_k(I_1), K_x))$ (Hash of ciphertext for string K_x and the public key of intermediator I_1.)

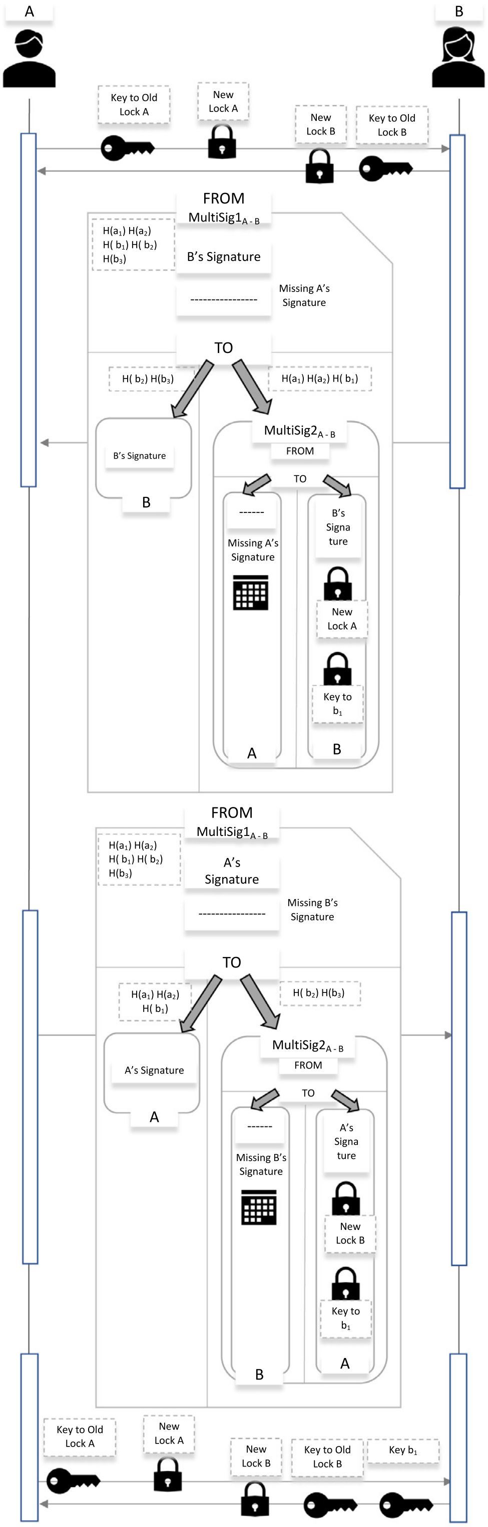

Fig. 7. Protocol for updating channel to exchange local serial numbers.

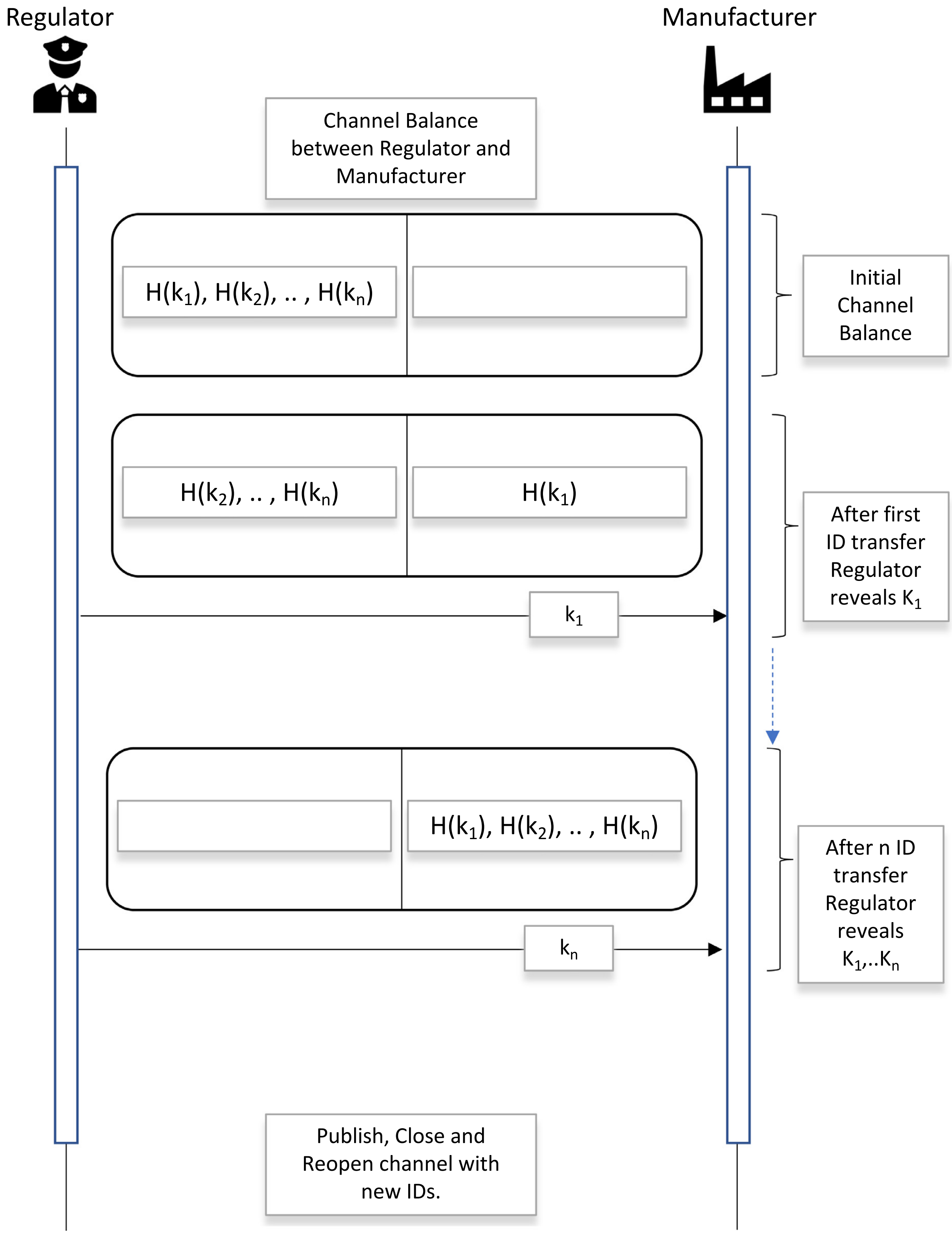
Regulator
Manufacturer
Channel Balance between Regulator and Manufacturer
$H(k_1), H(k_2), .. , H(k_n)$
Initial Channel Balance
$H(k_2), .. , H(k_n)$
$H(k_1)$
k_1
After first ID transfer Regulator reveals K_1
$H(k_1), H(k_2), .. , H(k_n)$
k_n
After n ID transfer Regulator reveals $K_1,..K_n$
Publish, Close and Reopen channel with new IDs.

Fig. 8. Protocol for secure ID transfer from regulator to the manufacturer.

- The manufacturer informs I_1 about $E(P_k(I_1), K_x)$ as I_1 can decrypt it with its private key to produce its Hash to satisfy the HTLC lock requirement.
- The time mentioned in each HTLC is less than the shelf life of the product and less than the time in the previous contract.
- HTLCs are formed as the product moves through the supply chain and it is only triggered by either the retailer or the consumer (described in the next section).

The above sequence of HTLCs can have the following outcomes of executions:

- The retailer or the consumer triggers the sequence of HTLCs before the shelf life of the product expires by sending $H(K_1)$ to all intermediator nodes using Onion routing. In this example, the encrypted message from the retailer to the intermediator I_2 is:

  $$E(P_k(I_2, H(K_1), I_2, E(P_k(I_1, H(K_1), Manufacturer))$$

  By decrypting it, I_2 will know that the key to the HTLC is $H(K_1)$ and the remaining message should be sent to I_1. Successful execution of HTLCs before the time mentioned in each HTLC will mean that the product is sold before its expiry date.
- If the HTLCs are not executed using $H(K_1)$ then it will be executed using the timeout option. It will indicate that either the product is not sold and it can not be sold in the future.

4.5. PBT execution by consumer

The protocol for serialization from the retailer to the consumer is as follows:

- The consumer pays for the product and receives $H(k_1)$ from the retailer.
- The consumer then informs the manufacturer about the identity of the consumer as its public key and label of the product $H(H(K_1))$.
- The manufacturer asks the consumer to establish an HTLC with the retailer with time out within next few minutes, the first Lock as $H(H(K_1))$ and second lock as $E(P_k(C), K_3)$ where $P_k(C)$ is the public key of the consumer and K_3 is a random string.
- This means the sequence of HTLC contracts for the product $H(H(K_1))$ is extended with a new HTLC contract.
- Next, the consumer triggers the HTLC contracts by producing $H(K_1)$ and $D(Priv_K(C), E(P_k(C), K_3))$ ($Priv_K(C)$ is the private key of the consumer).

4.6. Incentives for HTLC execution

In the previous sections, we used HTLCs to transfer local serial number between two parties. The incentive for executing PBTs in the Bitcoin Lightning network is to claim back funds. For example, assume that there are two HTLCs in a PBT as $HTLC_1$ between A and B and $HTLC_2$ between B and C. Here B is an intermediator node for a PBT between A and C. $HTLC_2$ is triggered by C as it produces the key to the lock of $HTLC_2$. B should use the same key to trigger $HTLC_1$. In crypto-currency networks, the incentive for B is to claim back tokens. In this case, B is claiming back tokens from A by trigging $HTLC_1$ which has already sent to C in $HTLC_2$. The incentives for B to trigger HTLCs in the PBTs mentioned in previous sections are as follows (Figs. 9 and 10):

- All parties are given fixed numbers to initial serial numbers and unspent serial numbers are required to create new serial numbers.
- Hence it is in the interest of node B to trigger the HTLC and claim local serial numbers from A, which it will use to create new local serial numbers.

4.7. Physical labelling of products

Products in the supply chain will have physical labelling $H(H(K_1))$ where K_1 is the actual serial number only known to the regulator and the manufacturer. Note that, after receiving a purchase order from the retailer, the manufacturer forms the sequence of HTLCs. As the product moves in through the intermediators, the product label is recorded and relevant events are recorded separately. These events are needed before the execution of the PBT. The retailer only execute the PBT after receiving the product and intermediators can not record product movement event without physically accessing the product, i.e., reading RFID tags in close proximity.

4.8. Offline network topology

The proposed solution is designed for a large scale supply chain where no party has absolute control over the supply chain and parties have various levels of trust issues with other members of the supply chain. We use offline channel networks similar to the Lightning network of Bitcoin. Besides the members of the supply chain, i.e., regulator, manufacturer, intermediators, retailer, consumers, there may be other permissioned nodes who can facilitate serialisations.

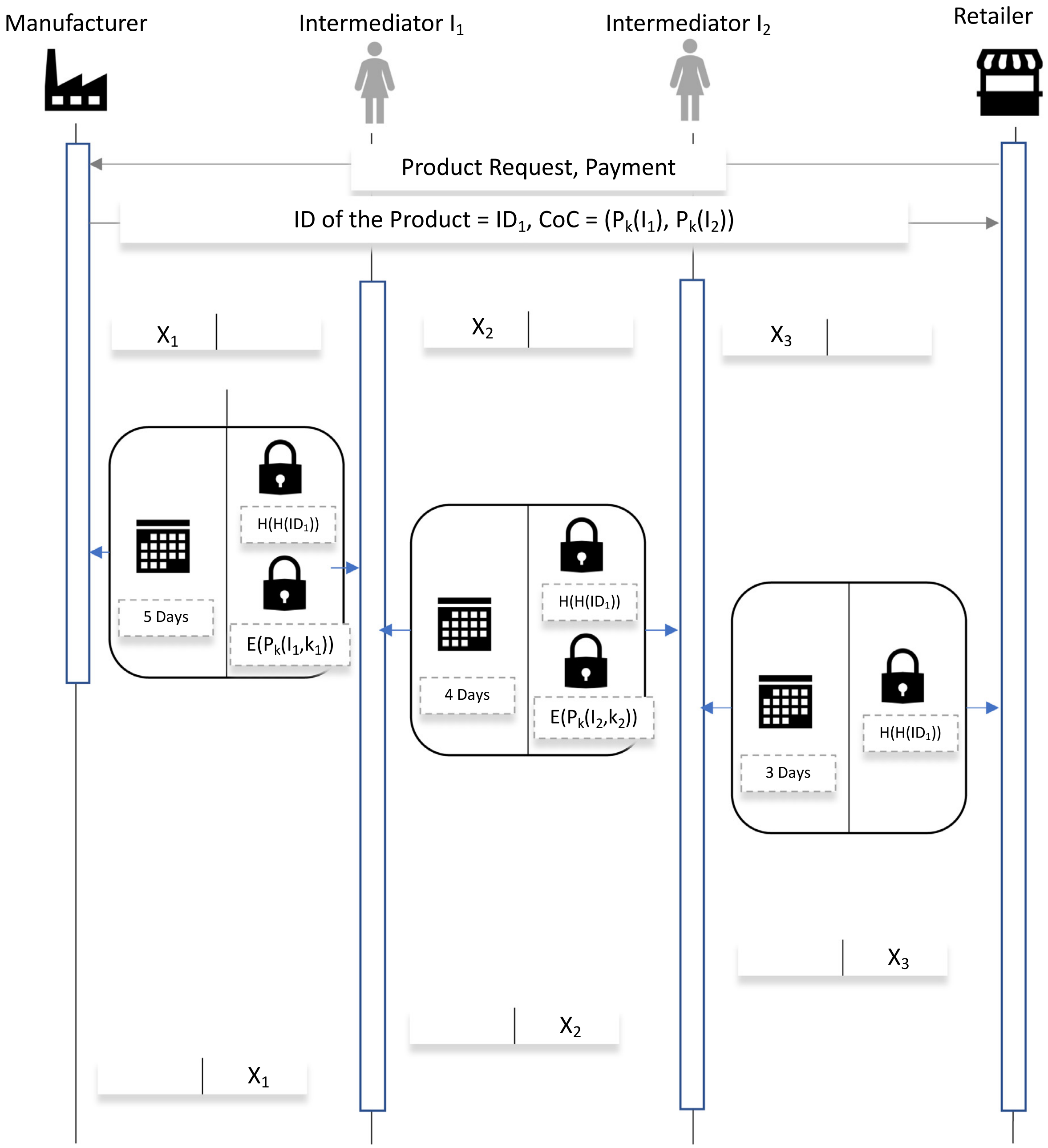

Fig. 9. Protocol for serialisation from manufacturer to the retailer.

4.9. Scalability

Our solution improves the scalability of the serialisation by using offline channels. Offline channels improve scalability by reducing the number of transactions needed to be recorded in the blockchain. There are only two transactions needed to be recorded for every channel. One transaction is needed to open the channel and one transaction is needed to close the channel. We use channels to transfer local serial numbers from one party to another. We allow a limited number of serial numbers to be used in one channel. Hence channels are regularly closed (i.e., change of serial number assignment is recorded in the blockchain). If in a channel a party can transfer x serial numbers to another party then reduction in number of transaction is $(x-2)/x$.

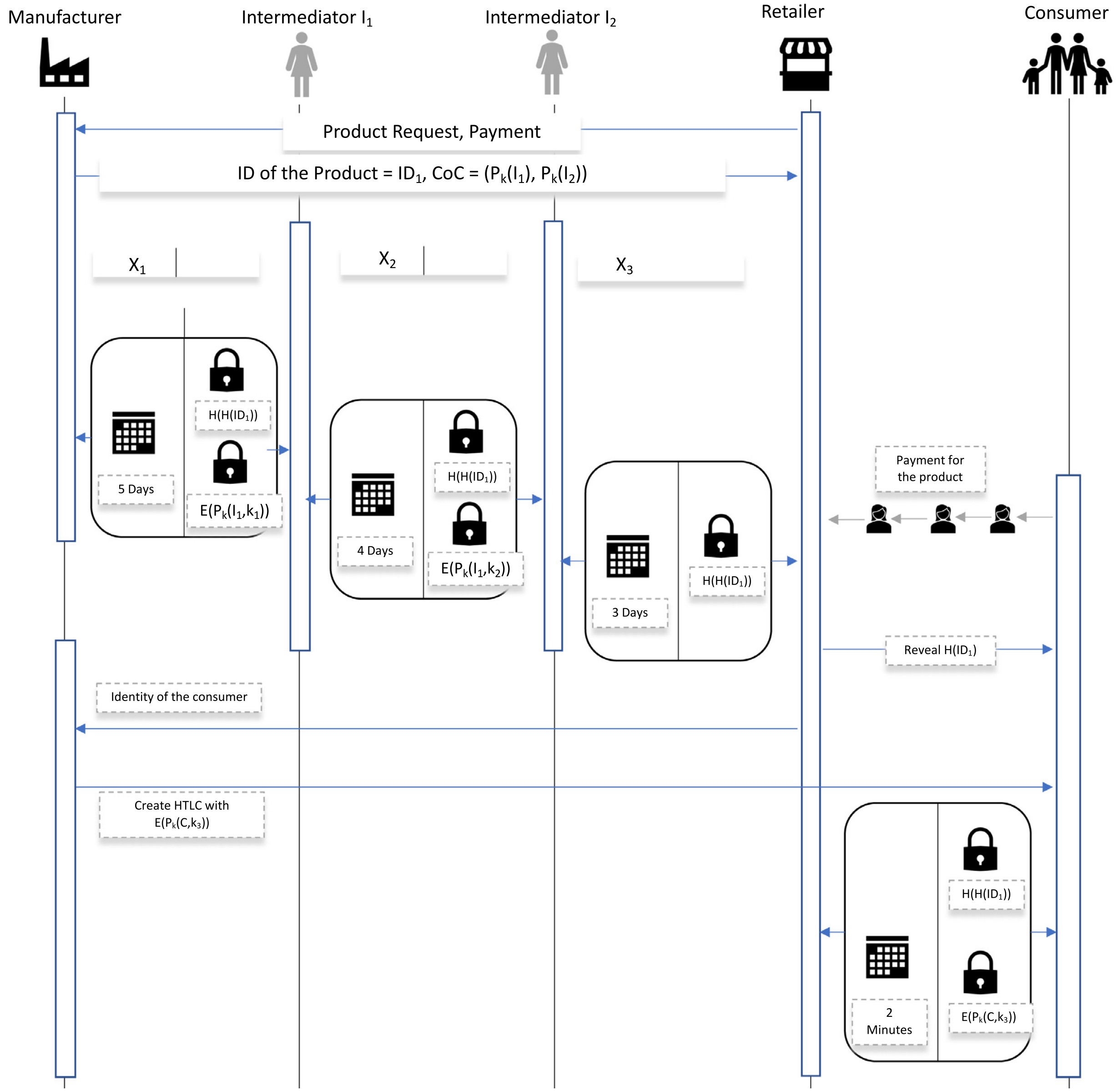

Fig. 10. Protocol for using offline channels.

5. Analysis

Fig. 11 illustrates the sequence of events as a serial number propagates through a supply chain according to our serialisation method. At t_1 the regulator records (in the blockchain and visible to all members of the blockchain) $H(H(H(K_1)))$ as a local serial number using the procedure mentioned in Section 4. At t_2, the regulator sends serial number K_1 to the manufacturer using an offline channel between them as mentioned in Section 4. At t_3, the retailer places an order for the product and receives $H(k_1)$ as the serial number. The manufacturer chooses a planned change of custody of the product through intermediator nodes and creates a sequence of HTLCs. In this example, there is one intermediator. At t_4 the intermediator knows about $H(H(K_1))$ as Lock of the HTLC. At t_5 the retailer knows about $H(H(K_1))$ as Lock of the HTLC. At time t_6 the consumer buys the product and gets the HTLC with lock $H(H(K_1))$ and key $H(K_1)$ (at time t_7). The consumer triggers the chain of HTLCs as the intermediator and the manufacturer receive $H(K_1)$ at t_8 and t_9 respectively. We will use this sequence of events to describe the attack scenarios.

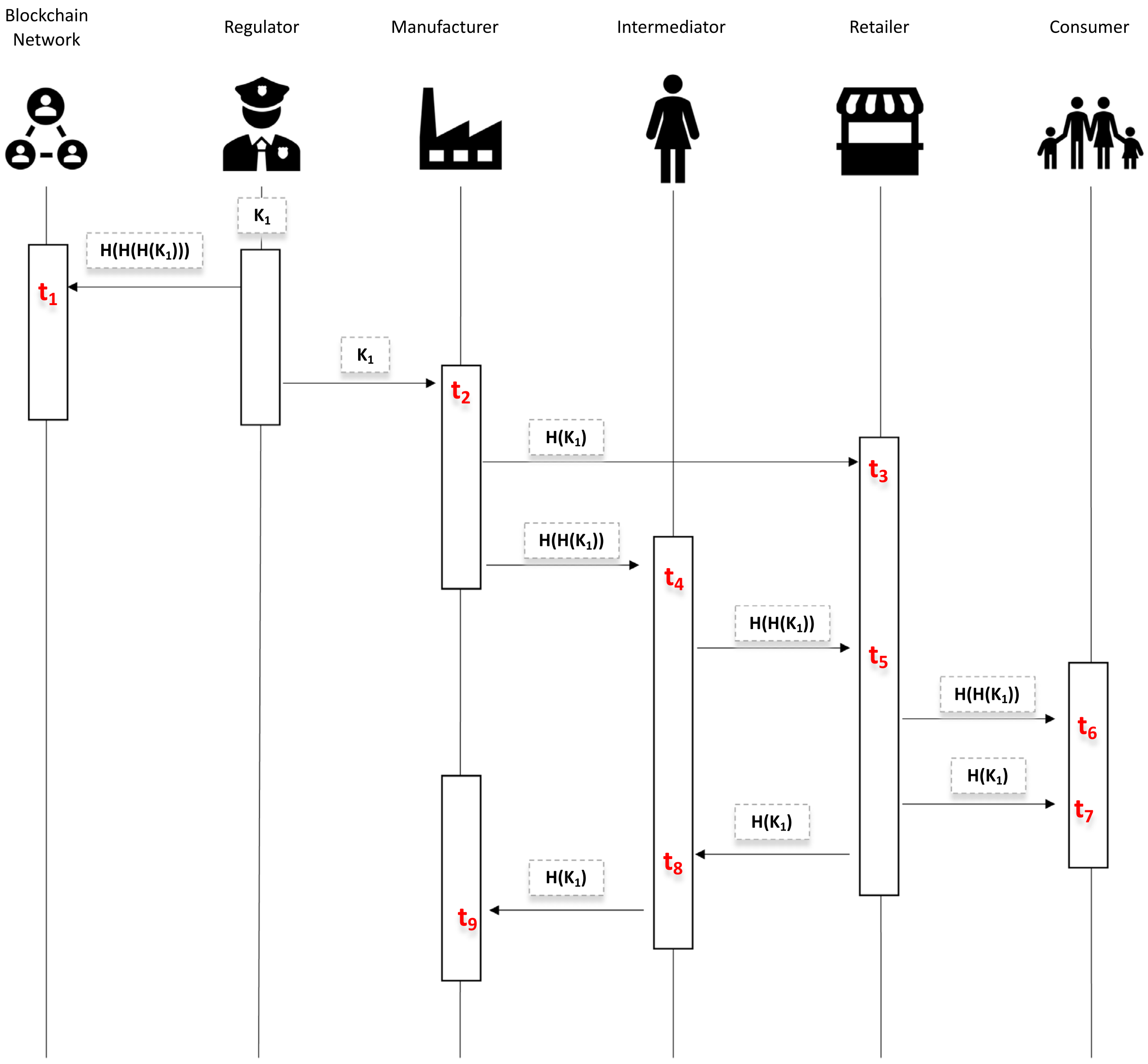

Fig. 11. Timeline of serial number propagation in the supply chain. H(x) is the Hash of x. t_i indicates the time of an event.

5.1. Solution to trust problem 1

Stealing serial numbers: Uniqueness of serial numbers is a concern for the regulator. In this case, an adversary steals a genuine serial number and label counterfeit products with it. Serial numbers can be stolen in the following scenarios:

- At t_1: The manufacturer records a valid serial number K_1 as $H(H(H(K_1)))$. All members of the blockchain network can access $H(H(H(K_1)))$ but it can not generate K_1 from it and hence no other party can label products using K_1.
- At t_2: The manufacturer receives the serial number at t_2 and it can label multiple products with the same serial number. In this case, there will be multiple PBTs to the consumer with the same lock for the HTLCs. As channels from manufacturer to consumer are regularly closed and reopened, transactions with the same lock for HTLC will be recorded in the blockchain. The regulator can periodically search the transaction list to find a duplicate lock for HTLCs. Hence the regulator can find that the manufacturer has labelled multiple products with the same serial number.
- At t_3: The retailer knows about the Hash of the serial number K_1. It is possible that the retailer reveals $H(K_1)$ to an adversarial manufacturer who wants to label a counterfeit product with $H(H(K_1))$ and create a similar sequence of HTLCs. In this case, if the sequence of HTLCs is executed then the blockchain will record multiple sequences of HTLCs with the same lock. The regulator can detect that by searching the transactions in the blockchain. A similar explanation holds if the consumer at t_6, t_7 reveals $H(K_1)$ to an adversarial manufacturer.

- At t_4: The intermediator has the information $H(H(K_1))$ as lock of the HTLCs and physical label of the product. It is possible to label a fake product with $H(H(K_1))$ but it is not possible to create the sequence of HTLCs as it is not possible to construct $H(K_1)$ from $H(H(K_1))$.
- At t_8: The intermediator knows $H(K_1)$ as the key to lock in its HTLC. It may share the key with an adversarial manufacturer but as mentioned before the regulator can search the transactions for HTLCs with duplicate locks.

5.2. Solution to trust problem 2

- Change of custody: We claim that the product will be verified if it has propagated according to a planned change of custody because:
  - It is not possible to execute the HTLCs by parties who are not in the planned change of custody. Note that, we include an additional Lock in the HTLC and its key can be only decrypted using the private key of a party. Hence an illegal change of custody will not be possible as long as combinations of public-private keys are secure.
- The shelf life of perishable food products: The manufacturer can be assured that a product is either sold before its expiry date or discarded after its shelf life expires.
  - The manufacturer decides on the time of each HTLC and it can choose the time locks in such a way that in the last HTLC the time is at most the shelf life of the product. This means either the product can not be sold after its shelf life as HTLC will not be executed.
- Stealing serial numbers: The manufacturer may be concerned about stolen serial numbers. Serial number stealing scenarios from t_3 to t_9 are applicable in this case. As explained in trust problem 1, in these scenarios serial numbers can not be stolen.

5.3. Solution to trust problem 3

- Change of custody: The change of custody problem for an intermediator node is to ensure that it receives a product and sends a product according to a correct change of custody. The intermediator can be assured of proper change of custody because:
  - Only a manufacturer decides on a planned change of custody and it communicates with intermediator as proper HTLCs are created.
  - It is possible that an adversarial manufacturer creates a change of custody with stolen serial numbers. In such a case, as mentioned in trust problem 1, the regulator can search for duplicate locks in HTLCs.
- The shelf life of perishable food products: The intermediator can ensure that it received a product before its shelf life expires because:
  - If an intermediator receives a product after the shelf life expires then it will mean that time in its HTLC has also expired. Hence it can not receive and send a product with expired shelf life.

5.4. Solution to trust problem 4

- The change of custody: We claim that the retailer can verify that a product will be verified if it has propagated according to a planned change of custody:
  - The retailer can initiate the execution of the sequence HTLCs by providing $H(K_1)$ to the intermediator node. It means that HTLCs with the lock $H(H(K_1))$ will be executed and hence change of custody for the product labelled with $H(H(K_1))$ occurred.
  - The retailer decides on additional key x_1 for each intermediator where the lock is $E_{PK_{I_1}}(x_1)$ (x_1 encrypted public key of intermediator I_1). Only I_1 can decrypt the additional lock in its HTLC.
- The shelf life of perishable food products: The retailer can ensure that it receives a product before its shelf life expires.
  - Same explanation as trust problem 3.

5.5. Solution to trust problem 5

- Change of custody: The consumer can verify that a product has propagated according to a planned change of custody because:
  - As the product is sold, the retailer informs the consumer $H(K_1)$ and forms an additional HTLC where $H(H(K_1))$ is the lock. The consumer can trigger the sequence of HTLCs and the successful execution of the HTLCs will indicate that proper change of custody is followed.
- The shelf life of perishable food products: The consumer can ensure it buys a product before its shelf life is expired because:
  - The time in the sequence of HTLCs from the manufacturer to the retailer is such that maximum time is less than the shelf life of the product. Hence successful execution of the HTLCs means that the product is sold before its shelf life has expired.

Table 1
A comparison with state of art.

	Security of serial numbers	*Secure change of custody*	*Control over perishable goods*	*Multi-party supply chains*	*Scalability*
[8]	Not solved.	Not solved.	RFID and other sensors are used.	Not solved.	Not solved.
[9]	Not solved.	Not solved.	Not solved.	Not solved.	Not solved.
[12]	Not solved.	Solved.	Not solved.	Not solved.	Uses G-Coin Blockchains.
[13]	Not solved.	Not solved.	Not solved.	Not solved.	Uses Hyperledger.
[22]	Not solved.	Not solved.	Solved.	Solved.	Not solved.
[23]	Not solved.	Solved.	Not solved.	Not solved.	Uses Hyperledger.
[24]	Not solved.	Solved.	Not solved.	Not solved.	Uses Hyperledger.
[25]	Not solved.	Solved.	Integrated with IoT.	Not solved.	Not solved (uses Ethereum).
[26]	Not solved.	Solved.	Not solved.	Not solved.	Not solved (uses Ethereum)
[27,28]	Not solved.	Solved.	Not solved.	Not solved.	Not solved (uses Ethereum)
[29]	Not solved.	Solved.	Not solved.	Not solved.	Uses Hyperledger Fabric.
[23,30]	Not solved.	Solved.	Not solved.	Not solved.	Uses Hyperledger Sawtooth.
Our solution	Solved.	Solved.	Solved.	Solved.	Solved.

6. Related literature

Archa et al. [7] uses blockchains for keeping immutable records of pharmaceutical product data. Feng Tian [8] uses blockchains to ensure traceability in food supply chains. Feng Tian [9] uses blockchains to ensure compliance with food safety regulations. Francisco and Swanson [10] investigates Unified Theory of Acceptance and Use of Technology to analyze the adoption of blockchain technology for supply chain traceability. Tse et al. [11] uses blockchain technology to ensure food safety in food supply chains. Tseng et al. [12] uses blockchains for ensuring governance in pharmaceutical supply chains. Kamath [13] presents the case study on Walmart's blockchain usage in its food supply chain. Mao et al. [5] presents blockchain-based traceability solutions in a multi-party supply chain. Galvez et al. [14] discusses the current challenges of blockchain managed food supply chains and these challenges include the security and scalability problems studied in this paper. Biswas et al. [15] discusses blockchain-enabled wine supply chains.

Offline channels are designed to improve the scalability of blockchains. Examples of such developments are as follows: Bitcoin Lightning network was proposed in [6] which allows peers to create and transfer funds among them without frequently updating the blockchain. Similar networks are proposed for Ethereum [16] and credit networks [17]. A privacy-preserving payment method in the credit network was proposed in [18]. Recent advances on the offline channel network are focused on the development of routing protocols for offline channels. Examples of such routing protocols are as follows: A method for anonymous payment to improve privacy in PBT was developed in [19]. Grunspan and Pérez-Marco [20] proposed a decentralised routing algorithm for the channel network.

In this paper, we investigate the product serialisation (and subsequent information management problems) problem for supply chains. Table 1 illustrates how our solution advances the state of art. We refer to [21] for a detailed review of blockchain-based traceability solutions for food supply chains. A summary of our contributions are as follows:

- **Scalability:** The current blockchain-based supply change management solutions may have scalability problems as all supply chain events are recorded in the blockchain. We reduce interaction with the blockchain by using offline channels. This improves the scalability of blockchain-managed supply chains.
- **Multi-party supply chains:** The current blockchain-based traceability solutions ignore multi-party supply chains where a single party has limited control over the supply chain.
- **Secure change of custody:** The current blockchain-based supply chain management solution may not ensure the proper change of custody of a product i.e., a product will move through the supply chain as planned by the manufacturer who does not control the distribution line.
- **Perishable food products:** The current blockchain-based supply chain management solutions do not enable the manufacturer to discard expired perishable food without control over supply chains.
- **Secure serial numbers:** The current blockchain-based supply chain management solutions do not ensure that serial numbers can not be used to label fake products.

7. Conclusion

In this paper, we developed a supply chain serialization method using blockchains. We used blockchain offline channels to improve scalability. We have shown that our solution adequately addresses the security challenges of product serialization. We proved that our method can prevent the sale of products with expired shelf life. In the future, we will investigate how to use this serialization method to automate product recall.

Declaration of Competing Interest

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

References

[1] GS1 standards and blockchain, 2019a, (https://www.gs1.org/standards/blockchaina), [Online; accessed 19-Nov-2019].

[2] 5 ways counterfeiting hurts society and what we can do about it, 2019b, (https://iccwbo.org/media-wall/news-speeches/5-ways-counterfeiting-hurts-society-and-what-we-can-do-about-it/b), [Online; accessed 19-Nov-2019].

[3] Its time to stop murder by counterfeit medicine, 2019c, (https://www.statnews.com/2019/05/07/stopping-murder-counterfeit-medicine/c), [Online; accessed 19-Nov-2019].

[4] N.O. Madichie, F.A. Yamoah, Revisiting the European horsemeat scandal: the role of power asymmetry in the food supply chain crisis, Thunderbird Int. Bus. Rev. 59 (6) (2017) 663–675.

[5] D. Mao, F. Wang, Z. Hao, H. Li, Credit evaluation system based on blockchain for multiple stakeholders in the food supply chain, Int. J. Environ. Res. Public Health 15 (8) (2018), doi:10.3390/ijerph15081627.

[6] J. Poon, T. Dryja, The bitcoin lightning network: scalable off-chain instant payments. https://lightning.network/lightning-network-paper.pdf

[7] Archa, B. Alangot, K. Achuthan, Trace and track: enhanced pharma supply chain infrastructure to prevent fraud, in: N. Kumar, A. Thakre (Eds.), Ubiquitous Communications and Network Computing, Springer International Publishing, Cham, 2018, pp. 189–195.

[8] Feng Tian, An agri-food supply chain traceability system for China based on RFID blockchain technology, in: 2016 13th International Conference on Service Systems and Service Management (ICSSSM), 2016, pp. 1–6, doi:10.1109/ICSSSM.2016.7538424.

[9] Feng Tian, A supply chain traceability system for food safety based on HACCP, blockchain & internet of things, in: 2017 International Conference on Service Systems and Service Management, 2017, pp. 1–6, doi:10.1109/ICSSSM.2017.7996119.

[10] K. Francisco, D. Swanson, The supply chain has no clothes: technology adoption of blockchain for supply chain transparency, Logistics 2 (1) (2018), doi:10.3390/logistics2010002.

[11] D. Tse, B. Zhang, Y. Yang, C. Cheng, H. Mu, Blockchain application in food supply information security, in: 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 2017, pp. 1357–1361, doi:10.1109/IEEM.2017.8290114.

[12] J.-H. Tseng, Y.-C. Liao, B. Chong, S.-w. Liao, Governance on the drug supply chain via gcoin blockchain, Int. J. Environ. Res. Public Health 15 (6) (2018), doi:10.3390/ijerph15061055.

[13] R. Kamath, Food traceability on blockchain: walmarts pork and mango pilots with IBM, J. Br. Blockchain Assoc. 1 (1) (2018), doi:10.31585/jbba-1-1-(10)2018.

[14] J.F. Galvez, J. Mejuto, J. Simal-Gandara, Future challenges on the use of blockchain for food traceability analysis, TrAC Trends Anal. Chem. 107 (2018) 222–232, doi:10.1016/j.trac.2018.08.011.

[15] K. Biswas, V. Muthukkumarasamy, W. Lum, Blockchain based wine supply chain traceability system, 2017.

[16] Raiden network, 2019, (http://raiden.network/), (accessed March 1, 2019).

[17] G. Malavolta, P. Moreno-Sanchez, A. Kate, M. Maffei, Silentwhispers: enforcing security and privacy in decentralized credit networks, IACR Cryptol. ePrint Arch. 2016 (2016) 1054.

[18] P. Moreno-Sanchez, A. Kate, M. Maffei, K. Pecina, Privacy preserving payments in credit networks: enabling trust with privacy in online marketplaces, NDSS, 2015.

[19] M. Green, I. Miers, Bolt: anonymous payment channels for decentralized currencies, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, in: CCS '17, ACM, New York, NY, USA, 2017, pp. 473–489, doi:10.1145/3133956.3134093.

[20] C. Grunspan, R. Pérez-Marco, Ant routing algorithm for the lightning network, CoRR (2018) abs/1807.00151.

[21] K. Demestichas, N. Peppes, T. Alexakis, E. Adamopoulou, Blockchain in agriculture traceability systems: areview, Appl. Sci. 10 (12) (2020), doi:10.3390/app10124113.

[22] A. Pal, K. Kant, Smart sensing, communication, and control in perishable food supply chain, ACM Trans. Sen. Netw. 16 (1) (2020), doi:10.1145/3360726.

[23] G. Baralla, A. Pinna, G. Corrias, Ensure traceability in european food supply chain by using a blockchain system, in: Proceedings of the 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain, in: WETSEB 19, IEEE Press, 2019, pp. 40–47, doi:10.1109/WETSEB.2019.00012.

[24] A. Iftekhar, X. Cui, M. Hassan, W. Afzal, Application of blockchain and internet of things to ensure tamper-proof data availability for food safety, J. Food Qual. 2020 (2020) 1–14, doi:10.1155/2020/5385207.

[25] M. Kim, B. Hilton, Z. Burks, J. Reyes, Integrating blockchain, smart contract-tokens, and IoT to design a food traceability solution, in: 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2018, pp. 335–340.

[26] G. Baralla, S. Ibba, M. Marchesi, R. Tonelli, S. Missineo, A blockchain based system to ensure transparency and reliability in food supply chain, in: G. Mencagli, D. B. Heras, V. Cardellini, E. Casalicchio, E. Jeannot, F. Wolf, A. Salis, C. Schifanella, R.R. Manumachu, L. Ricci, M. Beccuti, L. Antonelli, J.D. Garcia Sanchez, S.L. Scott (Eds.), Euro-Par 2018: Parallel Processing Workshops, Springer International Publishing, Cham, 2019, pp. 379–391.

[27] Q. Lin, H. Wang, X. Pei, J. Wang, Food safety traceability system based on blockchain and EPCIS, IEEE Access 7 (2019) 20698–20707.

[28] Y. Liao, K. Xu, Traceability system of agricultural product based on block-chain and application in tea quality safety management, J. Phys. Conf. Ser. 1288 (2019) 12062, doi:10.1088/1742-6596/1288/1/012062.

[29] D. Bumblauskas, A. Mann, B. Dugan, J. Rittmer, A blockchain use case in food distribution: do you know where your food has been? Int. J. Inf. Manage. 52 (2020) 102008, doi:10.1016/j.ijinfomgt.2019.09.004.

[30] K.Y. Chan, J. Abdullah, A.S. Khan, A framework for traceable and transparent supply chain management for agri-food sector in malaysia using blockchain technology, Int. J. Adv. Comput. Sci. Appl. 10 (11) (2019), doi:10.14569/IJACSA.2019.0101120.