Towards Privacy-Aware Traceability for Automotive Supply-Chains

DONGHANG LU, Purdue University
PEDRO MORENO-SANCHEZ, TU Wien
PRAMITA MITRA, Ford Motors
KEN FELDMAN, Ford Motors
JOSH FODALE, Ford Motors
JASON KOSOFSKY, Ford Motors
ANIKET KATE, Purdue University

Abstract:

The lack of traceability in today's supply chain system for auto components makes counterfeiting a significant problem leading to millions of dollars of lost revenue every year and putting the lives of customers at risk. Traditional solutions are usually built upon hardware such as RFID tags and barcodes and these solutions cannot stop attacks from supply chain (insider) parties themselves as they can simply duplicate products in their local database.

This industry-academia collaborative work studies the benefits and challenges associated with the use of distributed ledger (or blockchain) technology towards preventing counterfeiting in the presence of malicious supply chain parties. We illustrate that the provision of a distributed and append-only ledger jointly governed by supply chain parties themselves makes permissioned blockchains such as Hyperledger Fabric a promising approach towards mitigating counterfeiting. Meanwhile, we demonstrate that the privacy of supply chain parties can be preserved as competing supply chain parties strive to protect their businesses from the prying eyes of competitors and counterparties. Besides, we show that the recall process can be achieved efficiently with the help of the blockchain. The proposed solution, Fordchain, overcomes the challenges to achieve the best of both worlds: a solution to the counterfeiting problem using distributed ledger technology while providing accountability and the privacy notions of interest for supply chain parties. Although our efforts to build a blockchain-based counterfeiting prevention system aims at automotive supply chains, the lessons learned are highly applicable to other supply chains. We end-to-end implement our Fordchain solution in the Hyperledger Fabric framework, analyze it over AWS EC2 clusters, and illustrate that the performance of our solution is good enough to be applied in practice.

1 INTRODUCTION

The U.S. automobile industry is a significant part of the nation's economy, with the sales of auto components being a rapidly growing market. However, the huge supply chain system behind them makes the auto component an attractive target for the counterfeiting and grey-market. What's more, the customers may encounter life-threatening situations due to the low quality, durability and safety of counterfeit auto components. As a result, deterring counterfeiting of goods has become a key challenge for automotive companies. Therefore, the life cycle traceability of automotive components should be built up.

Barcodes as well as hardware solutions like physically unclonable functions (PUF) [33] and RFID [34] help in identifying and authenticating the goods to different degrees; however, they cannot prevent the counterfeiting from insiders (i.e., players who are part of the supply chain themselves). The adversarial supply chain players can easily equivocate (and modify the supply chain logs) to present conflicting views to other players and to the end-consumers proactively in the field, as well as reactively during security audits. The malicious behavior by supply chain entities has become a serious risk. For instance, in the now-famous horse meat scandal [4] in parts of Europe, some beef-based products were found to contain undeclared or improperly declared horse meat up to 100% of the meat content. The counterfeiting problem was hard to resolve due to the breakdown in the traceability of the food supply chain. It took a significant amount of manual efforts to track back the source of

horse meat, a trader called Draap Trading. Such scandals clearly demonstrate how much damage an entity inside supply chains can cause due to the absence of immutable logs.

Blockchains are emerging as a promising solution towards dealing with such malicious insiders. Blockchains [7, 11, 24, 31, 35, 36] form a distributed source of shared truth for the supply chain, which along with smart contracts and cryptographic primitives help mutually distrusting sets of players/companies with possibly adversarial interests to collaborate with a secure set of rules. The immutable, append-only blockchain ledger for the supply-chain stops incorrect data entries as well as later manipulations and thus allows the participants to verify and audit transactions. It is initially used as payment platforms for cryptocurrencies like Bitcoin [24]. With the appearance of Ethereum [36] and Hyperledger Fabric [7], the conception of smart contracts is proposed and can be used to enforce business logic through programming and without human interaction. As a result, several applications are built based on the blockchain when people see the potential of decentralization, transparency, and immutability [27, 38]; e.g., Walmart has been working with IBM to building a blockchain for food tracing and safety [16] . Recently, the auto industry started the mobility open blockchain initiative (Mobi) [1] towards using blockchain and related technologies to make mobility safer, greener, cheaper, and more accessible.

Amidst this enthusiasm in the industry, there are several unattended privacy concerns, which demand immediate attention. Adding transparency without understanding and addressing introduced privacy concerns can lead to leakage of trade secrets and intellectual property of the supply-chain parties. Indeed, some concerns [3] about this are already getting raised in these directions. This work focuses on understanding some of the fundamental security and privacy concerns associated with developing blockchain-based traceability solutions for automotive supply chains and resolving those by using cryptographic solutions.

Recently, in a position paper, Lu et al. [20] offered a high-level overview of the integration of supply chains and permissioned blockchain; In this work, we take several necessary steps further towards making blockchains practically relevant for the automatic supply chain. In particular, we observe and solve nuance challenges with respect to privacy and authentication expectations, recall procedures, and accountability measures. Moreover, we integrate a realistic supply-chain example in our general framework and develop the prototype to demonstrate how privacy, authentication, and accountability can be achieved through the blockchain and smart contracts.

1.1 Contribution

We take a simplified-yet-expressive supply-chain example of airbags from the automotive industry and develop an end-to-end blockchain instantiation, Fordchain, over the Hyperledger Fabric. Fordchain adds the identity and transfer of genuine components to the ledger at each step by the appropriate supply chain player, thus enabling traceability through the entire product lifecycle—starting from the components assembly at sub-tiers and Tier 1, followed by a final assembly at the original equipment manufacturer (OEM), and finally after sales and in the field until the components (or the vehicle itself) retires or gets scrapped.

We assume the vehicle to be "blockchain-aware" [20] in the sense that the car is capable of querying the blockchain ledger; thus, the vehicles can tell the owner about the validity of the current airbags. Besides, blockchain-aware vehicles are necessary to maintain aftermarket traceability and to resolve the recall process.

To combine the blockchain and supply chain properly, we study and provide an overview of the general model for the combination of supply chain and blockchain. To fit the solution with auto supply chains, we come up with a novel architecture design to achieve privacy-preserving airbag life-cycle queries and scalable product transfer. We developed a 1-peer-1-chaincode structure to extend the scalability of the system and support flexible access control. Privacy protection is a significant concern since a significant amount of commercially sensitive data is involved in supply chains. Privacy goals such as confidentiality are not trivial to achieve when the blockchain is designed to maintain a ledger where all actions are traceable [5, 14, 37]. Besides, we summarize a set of general principles regarding applying Hyperledger Fabric to supply chains.

Finally, we implement an end-to-end prototype using Hyperledger Fabric v1.4, and perform an evaluation of the prototype on AWS EC2 clusters for real-world supply-chain system loads. We developed code to deploy Hyperledger Fabric on multiple cloud servers automatically through Docker Swarm. We also set up Raft consensus clusters on multiple AWS servers to provide ordering service for our prototype. We observe that the Fordchain design is practical to use for real-world supply chains. The benchmark illustrates that Fordchain can handle the daily transactions of a supplier in minutes.

Paper Organization. We present the background of the auto supply chains and blockchains in Section 2. In Section 3, we formally describe a model of a typical supply chain and then specialize it for the automotive supply chains. We present our detailed construction of Fordchain based Hyperledger Fabric in Section 4. Section 5 shows the evaluation of Fordchain prototype. Section 6 contains relevant work and we conclude the paper in Section 7.

2 BACKGROUND

2.1 Automotive Supply Chain and the Issue of Counterfeiting

As a representative scenario, we consider the process flow of a safety-critical part, e.g., an airbag. The participants included in the life cycle of airbags are original equipment manufacturer (OEM), dealers, tier 1 suppliers, and multiple sub-tiers [22]. A valid airbag module consists of three major components which are the airbag module itself, the inflator, and the breakaway plastic horn pad cover. These three components are produced and assembled by various sub-tiers (tier 2 or 3) and the final assembly is done at tier 1, where the airbag module is given a unique serial number. Then the assembled airbags are shipped from tier 1 suppliers to the OEM, who can mount the airbags to new vehicles or distribute the airbags to official dealers. If a recall process starts, the customer is notified if its vehicles are included in the recall list, and they can go to any official dealers to replace the airbag.

The counterfeiting problem is the key issue that we aim at resolving and counterfeits can enter the supply chain system in multiple ways. For instance, the end user (i.e. the owner of a vehicle) can choose to replace airbags in unauthorized independent shops. It is often the case that small-scale shops choose cheap parts to maximize their profits and end up serving counterfeit parts [12].

According to previous works [20], quality teams at OEM itself (or at OEM and tier 1 supplier) work together to identify the counterfeiting issue. To that end, OEM first tries to identify if the part being inspected has the right OEM branding. A subset of counterfeits, especially for the OEM unique parts, could be detected in this method. On the other hand, for parts that are not OEM specific but rather are sold as a "black box" by Tier 1, detecting counterfeit involves asking Tier 1 to provide verification. The entire process is fairly manual and incurs less than optimal cost and latency, as the traceability is fragmented across the tracking systems of various supply chain players. Also, for "black box" parts where Tier 1 owns the intellectual property, the cause of the counterfeiting is not always shared seamlessly with the OEM. For instance, for "black box" parts, OEM may not have full visibility if one of the components (e.g., inflator of an airbag) manufactured by sub-tiers is the source of the counterfeits. Lack of a single, shared source of part traceability through its lifecycle, from sub-tiers all the way to aftermarket, prevents the OEM from performing adequate and timely risk analysis and mitigation strategies.

2.2 Distributed Ledgers and Blockchains

Distributed ledgers, with the most famous example being the blockchains [7, 11, 24, 31, 35, 36], are append-only databases maintaining consistency among a group of peers. The data are stored in the form of blocks, where each block includes a cryptographic hash of the previous block. Due to the append-only property, blockchain can resist others from modifying the history of the data. As long as recorded, the data block cannot be altered without the alteration of all subsequent blocks. Meanwhile, blockchains provide higher availability since blockchains are distributed systems where we do not need to trust any single party to maintain the database.

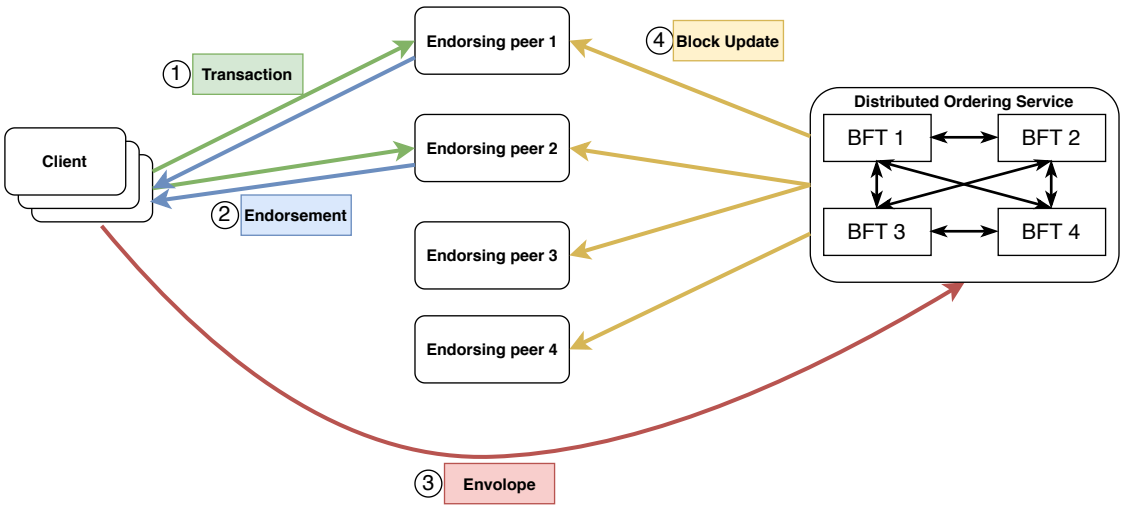

Fig. 1. Transaction Flow of Hyperledger Fabric. The example includes multiple clients, four peers who play as both endorsing peers and validating peers, and a distributed ordering service consisting of four ordering nodes running BFT consensus protocols.

Generally, blockchains are divided into two categories: permissioned blockchains and permissionless blockchains [13]. A permissionless blockchain [24, 36] does not put requirements on which users can interact with the network, submit transactions, and maintain the ledger. On the other hand, permissioned blockchain is a system where the identities of participants are known to each other. Therefore, permissioned blockchains are often a better choice for enterprise users on internal business operations.

In this work, we consider automotive supply chains. The structure of the automotive supply chain can be visualized by a pyramid where the original equipment manufacturer (OEM) stands at the top and multiple tiers of suppliers stand layer by layer at the bottom. Meanwhile, the dealers directly connect with OEM and have no connection with any suppliers. We pick permissioned blockchain to fit the automotive supply chain as the identities of the network are known and fixed.

Hyperledger Fabric. We consider Hyperledger Fabric [7] in this work. Hyperledger Fabric is an open-source permissioned blockchain framework supporting modular and configurable architecture. The identities of participants in Hyperledger Fabric are known to each other, although parties may not trust each other. The blockchain remains consistent on all parties due to the use of consensus protocols.

Smart contracts are supported by Hyperledger Fabric in the form of chaincodes. The chaincodes are used to verify the validation of transactions, and they are stored and executed by endorsing peers, who maintain the blockchain (the ledger). Other roles in Fabric include the clients who send transactions to endorsing peers for validation (called endorsements), ordering nodes who run consensus protocols, and validating peers who validate transactions after ordering. The endorsing peers and validating peers can be the same group of parties, but not necessary.

Below is a general transaction flow of Hyperledger Fabric summarized by Lu et.al. [20] and each step is visualized in Fig. 1.

(1) A client generates a transaction and sends it to endorsing peers for endorsements. A transaction contains information such as the chaincode name, channel name, parameter field, client's signature and

some optional fields like transient fields. This information is required because several chaincodes are simultaneously supported. Moreover, each chaincode specifies an *endorsement policy* that defines which endorsing peers must receive this transaction for a valid endorsement.

(2) Endorsing peers, upon reception of a client transaction, first check if the transaction is well-formatted and if the client is authorized to perform the transaction. Then, endorsing peers will execute the transaction and generate a read set and write set containing the result of the execution together with endorsing peers' signatures. Importantly, at this point, the state of the ledger is not modified yet. The transaction is just being "simulated" to generate the expected output.
(3) Clients eventually receive the endorsement from the endorsing peers, check its content, attached signature and if all endorsements have consistent read set and write set. Then, when clients receive enough of such endorsements (defined by the endorsement policy), it will combine them together to form an envelope and send it to the ordering service.
(4) The ordering service is carried out by a set of nodes (possibly different from the endorsing peers) that execute a consensus protocol to agree on the order of transactions, independently of their content. The sorted list of transactions is then included in blocks which are finally sent to the validating peers.
(5) Validating peers will validate the transaction inside the envelope again to confirm that the endorsement policies are satisfied and that the current state of the blockchain is consistent with reading set in the envelope. Once all checks pass, peers will apply the change to their current state and append the received block to the blockchain.

3 MODELING AUTO SUPPLY CHAINS

In this section, we overview our modeling of the supply chain for automobiles.

3.1 Supply Chain Model

Our general model for supply chains is a distributed system that must provide traceability of objects among a mutually untrusted set of players. We first outline the high-level architecture and principles of operations, then cover the considered threat model, and discuss the required security goals.

3.1.1 Architecture and Principles of Operation. The supply chain's conceptual architecture consists of a set of entities $\mathcal{E}$ and a set of objects O.

Objects. Each object $o \in O$ represents a traceable item within the supply chain. Each object has associated a unique ID, which could be used to identify the object with the help of hardware such as PUF and RFID, and a *state* that define its current status (e.g., mounted or retired for an airbag). Moreover, objects can be composed of a set of one or more *components*. For instance, an airbag consists of three components: fabric bag module, the inflator and the plastic horn. Finally, we say that an object is *owned* by a certain entity if such entity received the object from another entity and has not transferred it forward in the chain yet.

Entities. Each entity $e \in \mathcal{E}$ represents a party involved in the production and distribution of objects within their lifetime in the supply chain. We classify entities in the supply chain depending on their role as follows:

- *Entry Node*: An entry node introduces components to the supply chain for the first time. In an end-to-end solution, an entry node is the generator of the components.
- *Assemble Node*: Assemble nodes provide assemble services to combine components to form new objects
- *Relay Node*: A relay node is an entity in charge of transferring objects in the supply chain across other entities.
- *Exit Node*: An object ends its life cycle within the supply chain on an exit node, that is, after an object has been processed by an exit node, the object goes to "aftermarket" where it is serviced and ultimately

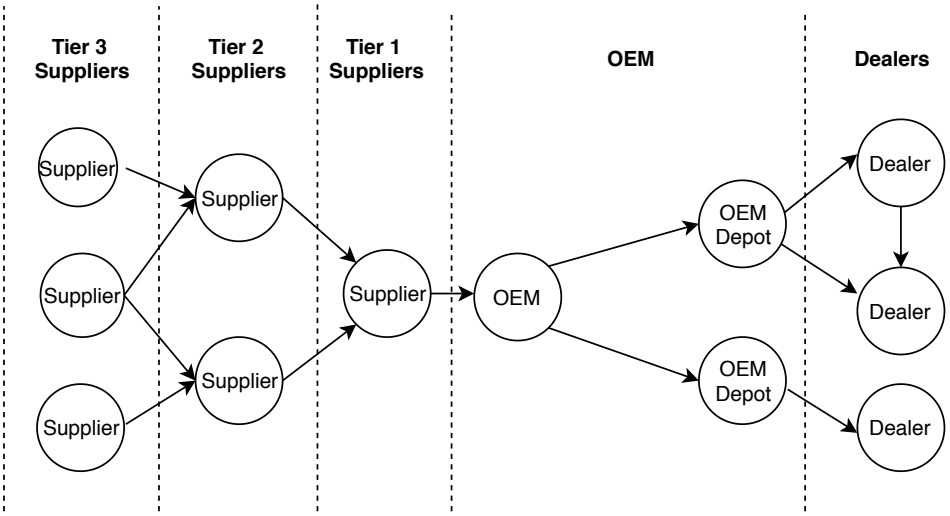

Fig. 2. System model of the auto supply chain

scrapped/retired at the end of life. The customer can get an equivalent object (e.g., a service part) at any exit node (e.g., dealership) no matter which exit node the customer buys products from.

- *Central Node*: We describe central nodes as nodes where all products must go through. Central nodes exist in most supply chain use cases and could be used to manage information of all products.
- *Client Node*: Client Nodes represent the customers who buy objects from the supply chain network. They are not parts of the set of entities that transfer objects from producers to consumers, but rather query information from them to know the state of an object at each moment.

3.2 Auto Supply Chain Instantiation

An example of auto supply chains is illustrated in Fig. 2. Similar to the general supply chain case, we consider an auto supply chain network consisting of a set of entities $\mathcal{E}$ and a set of objects O defined as follows:

Objects. In auto supply chains we consider airbags, components which are parts of airbags, and cars as the objects that can be included in the supply chain. Each airbag consists of three components including a fabric bag, the inflator, and the breakaway plastic pad cover. Each airbag is assigned a unique ID "xyz" which is the concatenation of three components' IDs "x", "y" and "z". Note that, for simplicity, we have restricted the type of objects and the number of components at each object. However, our model can be seamlessly extended to handle several types of objects, each of them can have several numbers of components.

Entities. Entities represent parties who participate in the life cycle of airbags. Below, we describe the considered entities and how they map to the roles in a supply chain:

- *Supplier.* An entity is a *supplier* if it is in charge of introducing legitimate objects into the supply chain. We differentiate between *tier 1* and *sub-tier* suppliers. A sub-tier supplier introduces legitimate components into the supply chain whereas a tier 1 supplier collects components from lower-tier suppliers and assemble them to construct airbags.
- *Manufacturer (OEM).* An entity in charge of mounting objects into the cars for the first time. Moreover, the manufacturer can recall a set of objects that have been detected faulty. OEM also keeps track of all

Table 1. Entity-Role mapping in auto supply chains.

Entity	Roles					
	Entry Node	Relay Node	Assemble Node	Central Node	Exit Node	Client
Sub-tier Supplier	●	●	○	○	○	○
Tier 1 Supplier	○	●	●	○	○	○
OEM	○	●	●	●	○	○
OEM depot	○	●	○	○	○	○
Dealer	○	●	○	○	●	○
Car	○	○	○	○	○	●

airbags to maintain traceability. Besides, OEM owns some OEM depots which distribute products to dealers for sale.
- *Dealer.* An entity that replaces objects included in a car that are either damaged or recalled by the corresponding manufacturer.
- *Car.* An entity that represents the car and plays the role of clients in supply chains.

Operations. We now introduce the operations that we consider in auto supply chains. In general, we consider the following operations:

- Add Component: It is used to add a component to the supply chain. For instance, each airbag consists of three components: the inflator, a horn pad cover and a fabric bag modular. The operation is rejected if the ID of the component already exists.
- Assemble Airbag: It combines different kinds of components to form an airbag. This operation takes three components as input, checks if all components are valid (i.e., all components belong to the entity and none of them are used.), then output an airbag.
- Transfer: It is used for transferring airbags or components. The object must belong to the sender, otherwise, the operation is rejected.
- Mount Airbag: It mounts a fresh airbag to a new car. The car has to be a new car and the airbag must be valid (it is a new airbag and it belongs to the entity).
- Recall Airbag: It recalls airbags with quality problems. OEM can issue such a recall, assigning recalled airbags to authorized dealers so that car owners can get their airbags replaced in corresponding dealers. The recalled airbags become retired and can never be reused in any case.
- Replace Airbag: It retires an old airbag in a car and installs a new valid airbag into the car. The new airbag must be valid.
- Buy Car: It transfers the ownership of a car to the customer.
- Check Airbag: It is used to check if the status of an airbag is valid (i.e., It is not retired or recalled).
- Historical/Forward Query: It is used to query the life cycle of an airbag or component.

We map the roles in car supply chain systems into thethe model above, and the mapping is shown in Table 1. Each role can perform one or more operations. The mapping is shown in Table 2. For example, we give central nodes (OEM) and the clients (car owners) the permission to do historical query such that they can trace the life cycle of the parts.

Table 2. Role-Operation mapping in auto supply chains.

Operation	Roles					
	Entry Node	Relay Node	Assemble Node	Central Node	Exit Node	Client
Add Component	●	○	○	○	○	○
Assemble Airbag	○	○	●	○	○	○
Transfer	○	●	○	○	○	○
Mount Airbag	○	○	○	●	○	○
Recall Airbag	○	○	○	●	○	○
Replace Airbag	○	○	○	○	●	○
Buy Car	○	○	○	○	○	●
Check Airbag	○	○	○	○	○	●
Historical/Forward Query	○	○	○	●	○	●

3.3 Threat Model and Security Goals

We consider two types of entities in the supply chain: *authorized* and *unauthorized* entities. Authorized entities are entities in a consortium for supply chain willing to respect the consortium rules and yet eager to learn the business from competitors. In extreme cases, however, they can behave maliciously trying to corrupt the whole system or break business rules for profits. On the other hand, unauthorized entities are considered fully malicious as they are not part of the consortium.

We assume that public key infrastructure (PKI) is established. All *authorized* entities are associated with valid public keys, however, *unauthorized* entities are not registered. We also assume that each component has a unique identifier that cannot be detached/removed from the component. It should not be possible for the adversary to create new valid identifiers. The adversary may however collect used identifiers from other components in the supply chain.

As for security goals, a supply chain should achieve the following security and privacy goals:

- *Confidentiality*: A supply chain must ensure that an entity does not learn any information from the objects that it has never owned. Note that an entity can derive certain information about an object that it has owned in the past. Yet, the supply chain must ensure that an entity does not learn additional information about a previously owned object after it transfers the object to another entity. Confidentiality ensures thus that every entity's business is kept private from the prying eyes of the competitors.
- *Authorization*: The supply chain must ensure that an entity is not able to perform an operation over an object that it does not own. Moreover, the supply chain must ensure that an entity can only perform the actions corresponding to its role. Authorization is thus crucial to ensure that each entity only has the right to carry out actions within its business scope.
- *Accountability*: The supply chain must ensure that if any entity does not follow the rules encoded in a certain action, a misbehavior proof can be then computed to blame the misbehaving entity. Moreover, the supply chain must ensure that a malicious entity cannot compute a misbehavior proof to falsely blame an honest entity.

4 FORDCHAIN: AUTO SUPPLY CHAIN OVER HYPERLEDGER FABRIC

In this section, we describe how we implemented our solution called Fordchain on Hyperledger Fabric 1.4 and the methods to achieve confidentiality, authorization, and accountability.

4.1 System Model

Each entity is represented as an endorsing peer of Hyperledger Fabric which maintains a copy of the blockchain and decides if transactions are valid by executing chaincodes. Besides, entities also play as clients who interact with blockchain by sending transactions. Moreover, the cars, as self-aware entities, play as clients and they can interact with the blockchain network by sending transactions to different endorsing peers.

As endorsing peer, each entity runs chaincodes including functions in the scope of its business (e.g. a sub-tier supplier includes AddComponent() into its chaincodes, while it does not include MountAirbag()). The mapping between entities and chaincodes they are supposed to run is shown in Fig. 5.

All actions happening in auto supply chain are abstracted to transactions. For instance, when a component is produced and added into the supply chain, the sub-tier supplier client will form a AddComponent() transaction and send it to sub-tier endorsing peer. If all checks pass the transaction will be endorsed by sub-tier endorsing peer and recorded in the blockchain as a valid transaction.

4.2 Detailed construction

4.2.1 System Assumptions. We assume all supply chain parties are responsible to maintain endorsing peers and keep these peers online to provide blockchain services. We also assume public key infrastructure is established so that the identities of parties are authorized. Besides, it is assumed that proper binding methods are applied so that each physical object is bounded with a digital ID, and this ID is used as the representation of objects in blockchain network.

4.2.2 Cryptographic Building Blocks. There are several cryptographic tools applied in Fordchain to achieve security and privacy goals.

Transport Layer Security [28] (TLS) are cryptographic protocols providing communication security over networks. It aims at privacy and data integrity and in Fordchain and we leverage it as a secure communication channel so that the communication between nodes is hidden from the adversary.

Due to the privacy requirements of Fordchain, some data on blockchain should be hidden. To achieve it we introduce symmetric encryption and cryptographic commitments into Fordchain. AES256-CBC [15] mode is used as the encryption scheme to hide transaction data on blockchain. Besides, commitments are used to keep data private meanwhile maintain accountability. In Fordchain prototype SHA2 [26] is used as the commitment method.

A digital signature is a scheme for verifying the authenticity of digital messages. In Fordchain, we leverage digital signatures as proofs and record signatures on the blockchain so that malicious parties cannot lie about their misbehavior. ECDSA signature scheme [18] is applied in Fordchain prototype.

4.2.3 Data Model. We implemented Fordchain prototype in Hyperledger 1.4. We define three data objects: components, airbags, and cars. Their structures are as follows:

- Components:
  - ID: A unique alpha-numeric string representing the component.
  - Status: current status of the component ("new", "mounted", "retired" or "transfered").
  - Previous Owner: previous owner of such component, if any.
  - Next Owner: next owner of such component, if any.
- Airbags:

  - ID: A unique ID representing the airbag, each airbag ID is the concatenation of three unique component IDs.
  - Status: current status of the airbag ("new", "mounted", "retired" or "transfered").
  - Previous Owner: previous owner of such airbag, if any.
  - Next Owner: next owner of such airbag, if any.
  - Location: the location where this airbag is mounted or replaced (e.g., some dealers).
- Cars:
  - ID: VIN number of the car.
  - Owner: owner of the car.
  - Airbag: current airbag mounted in this car.

Each sub-tier supplier peer has its components supply. Tier 1 suppliers, dealers, and OEM keep airbag supply where OEM airbag supply has records of all airbags in the system. As a result, any pair of airbags with duplicated component identities will be detected in OEM airbag supply. OEM also keeps all records of cars to issue recall efficiently.

4.3 Operations

In Fordchain, we make use of smart contracts, which are called chaincodes in Hyperledger Fabric, to realize business logic of different parties. Smart contracts collaborate to enforce that counterfeit products never enter the Fordchain system.

Fig. 3 and Fig. 4 illustrate the chaincode logic of all functions mentioned in Section 3.2. In each chaincode function, multiple checks are executed and the general goal is to reject transactions if input are duplicated IDs or retired IDs. There are more specific checks to make sure business logic is followed. Only when the transactions completely follow business rules will transactions be accepted and recorded into the blockchain. Besides, we keep the status for each airbag and component to make sure recalled or retired objects could not be reused, therefore preventing counterfeit objects from entering the supply chain by making use of valid IDs on retired objects.

For example, in the chaincode function *AddComponent*(), we reject the transaction if the input component already exists in the system. Endorsing peers also keep a list of retired components or recalled components to make sure they never enter the system again. By setting up these restrictions properly in chaincode, counterfeit products and retired products can never be added to the Fordchain system at entry nodes. Besides, the whole life cycles of valid components and airbags are traceable since relevant transactions are recorded in the blockchain. However, this method does not completely prevent counterfeit airbags from entering the system if the supplier itself is malicious. A malicious supplier can instantiate a malicious version of chaincode to skip these checks, add two components with the same id, then send two duplicated components to different higher tier suppliers without being caught immediately. However, this supplier will be finally caught when two components go to the central node (OEM), and all the proofs of misbehavior are stored in blockchain. The blockchain guarantees that the behavior of the malicious supplier is recorded by all honest parties, and cannot be modified. This is what the traditional databases with OEM as the central node cannot achieve.

Besides, we enforce every *BuyCar*() operation to be endorsed by OEM, such that OEM can have a global view of the ownership of all cars. This is essential to make the recall process efficient (Otherwise, ownership information of all cars has to be distributed among dealers, as a result, the recall process has to involve many parties). With our design, the OEM is the only party who can issue a recall, and the OEM alone has enough information to do so.

If a recall procedure is issued by OEM, the owners of the cars involved can go to any dealer to replace the buggy parts. And meanwhile, the car owner (or the “self-aware car" itself) will send two transactions $NotifyOEMofReplacement()$ and $NotifyDealerofReplacement$ to OEM and Dealers, such that the information can be updated.

Add Component

- Input: Component ID *id*
- Procedure:
  - Check if peer belongs to entity "Sub-tier Supplier".
  - Reject if component *id* exists or once existed in peer's supply.
  - If all requirements hold: add *id* into peer's supply, mark the status of *id* as new.

Receive

- Input: Product ID *id*, Sender identity *s*, sender signature *sig*.
- Procedure:
  - Reject if *id* already exists in peer's supply.
  - Check if the signature *sig* is valid, which means message is consistent with *id* and it is signed by sender *s*.
  - If all requirements hold: Add *id* into peer's supply. Update the previous owner of *id* to be *s* and store *sig* as proof of transfer.

Confirm

- Input: Product ID *id*, Receiver identity *r*, receiver signature *sig*.
- Procedure:
  - Reject if *id* does not exist in the record of peer's supply.
  - Check if the signature *sig* is valid, which means message is consistent with *id* and it is signed by receiver *r*.
  - If all requirements hold: Update the status of *id* to be "transfered". Update the next owner of *id* to be *r* and store *sig* as proof of transfer.

Assemble Airbag

- Input: Three Component ID $id1$, $id2$, $id3$
- Procedure:
  - Check if peer belongs to entity "Tier 1 Supplier".
  - Reject if any component does not exist in peer's supply.
  - Reject if status of any component is "mounted" or "retired".
  - If all requirements hold: create a new airbag ID *id_airbag*, which is the concatenation of $id1$, $id2$ and $id3$, then add *id_airbag* as a valid airbag ID into peer's supply, mark the status of $id1$, $id2$, $id3$ as new and mark $id1$, $id2$ and $id3$ as "assembled".

Mount Airbag

- Input:airbag ID *id*, VIN number VIN.
- Procedure:
  - Check if peer belongs to entity "OEM".
  - Reject if airbag *id* does not exist in peer's supply or the status of *id* is "mounted" or "retired".
  - Reject if VIN has existing record of mounted airbags.
  - If all requirements hold: update the owner of airbag *id* to be VIN. Update car supply as VIN.airbag = *id*.

Buy Car

- Input: VIN number VIN, Customer *c*.
- Procedure:
  - Check if peer belongs to "OEM".
  - Reject if VIN is not in OEM's car supply or the status of airbag mounted in VIN is "recalled".
  - If all requirements hold: Update the owner of VIN to *c* in OEM's car supply.

Fig. 3. Fordchain Chaincode Logic: Standard Supply-chain Operations

In what follows we discuss how we achieved confidentiality, authorization, and accountability in Fordchain.

Recall Airbag

- Input: VIN number VIN.
- Procedure:
  - Check if peer belongs to entity "OEM".
  - Get the corresponding airbag ID id which is mounted in VIN by querying car supply in OEM database.
  - Reject if the status of id is not mounted.
  - If all requirements hold: update the status of airbag id to be "recalled".

Notify OEM of Replacement

- Input: Airbag ID id, VIN number VIN, Dealer d.
- Procedure:
  - Check if peer belongs to "OEM".
  - Reject if VIN does not exist in OEM's car supply.
  - Reject if OEM has no record regarding id.
  - If all requirements hold:
    - Query OEM car supply to get old airbag ID in VIN to be id_{old}. Update the status of id_{old} to be retired on OEM database.
    - Update the status of id to be "mounted" and update $id.location = d$, $id.owner = VIN$.
    - Update OEM's car supply so that $VIN.airbag = id$.

Notify Dealer of Replacement

- Input: Airbag ID id.
- Procedure:
  - Check if peer belongs to "Dealer".
  - Reject if id does not exist in peer's supply or the status of id is not "new".
  - If all requirements hold: Update the status of id to be "mounted".

Fig. 4. Fordchain Chaincode Logic: Auto Supply-chain Specific Recall/Replace Operations

4.4 Approaches to Achieve Security Goals

4.4.1 Confidentiality in Fordchain. In Hyperledger Fabric all endorsing peers share the same ledger, thus share the same view of the blockchain and database. However, as we mentioned before, privacy protection (confidentiality) is a significant part of the supply chain and sensitive data should be hidden on the blockchain.

There are various options to keep data hidden such as encryption and commitments. We designed our solution for confidentiality by using the combination of these methods on transactions and the corresponding data. Intuitively, by doing that we can achieve different levels of confidentiality protection. In Fordchain, we propose to use two technologies to hide data: symmetric encryption and cryptographic commitments.

The reason that these two candidates are chosen instead of technologies like public-key encryption and zero-knowledge proof is that symmetric encryption and crypto commitments are light weighted and efficient to compute. Besides they are sufficient in the supply chain use case.

First, a client encrypts the transaction using symmetric encryption so that only the receiver could access the content of the transaction. Moreover, this ensures that the transaction is included in the blockchain in its encrypted form, so that other peers cannot see its content. Second, the endorser peer decrypts the transaction and parses it according to the chaincode instructions. The response message from the peer must be also hidden as it goes over other entities in the architecture (e.g., ordering service and other peers). Therefore, the endorsing peer commits his response so that it provides confidentiality. As a result, although blockchain is available to all endorsing peers, the contents are hidden so only the ones with the appropriate confidentiality level can see the plaintext of transactions and ledger changes. Note that this will not influence the transaction flow of Hyperledger

Fabric (e.g. transaction validation) because those steps do not rely on the plaintext value of transactions and endorsements. Besides, this method could be built up as an upper layer of Hyperledger Fabric, which means it is not necessary to change Hyperledger Fabric itself.

A transaction consists of many fields such as chaincode name, parameters, peers and so on. We can get a trade-off between efficiency and privacy by encrypting different fields based on the use case. It's often the case that the parameter field contains sensitive data so we propose parameter field should be encrypted to achieve minimum privacy. To achieve higher level of privacy, we can encrypt more fields such as the timestamp, blockchain identifier (so-called channel in Hyperledger Fabric), or channel information. To hide some of these fields, low-level code of Hyperledger Fabric may have to be changed since the decryption steps may be required before the execution of smart contracts.

An alternative method to hide data in the parameter field is to use "transient field". The transient field is a special field in Hyperledger Fabric with the property that the data inside will not be recorded on the blockchain. Thus it could be used to pass data that only endorsing peers can access. It is often treated as a channel to pass private data such as encryption keys. In the supply chain, data like product price or airbag ID are considered sensitive and the transient field could be used to keep these data from being recorded on the blockchain. However, accountability will be lost since nothing is left on blockchain if we leverage the transient field to pass input. As a result, it is required to put commitments of chaincode execution results on blockchain to maintain accountability.

Hyperledger Fabric provides a feature called private data collection starting from version 1.2, which is also an option for confidentiality protection. Compared to private data collection, the method we proposed is more flexible and users can encrypt different fields to find a balance between privacy and efficiency. Another option is to use trusted execution environment (TEE) such as Intel SGX to provide confidentiality [10]. Compared with this option, our solution does not rely on any hardware and does not suffer from performance downgrade caused by TEE.

Example of the Blockchain + Confidentiality. In what follows, we give a concrete example of the transaction flow when we take confidentiality into consideration. Suppose that a Tier-2 supplier wants to send a *AddComponent*() transaction, it first generates a symmetric encryption key k, then uses key k to encrypt the parameters of the transaction, which includes the function name "AddComponent()", and the component ID. Then it puts the encryption key k into the transient field of the transaction[1]. Finally, it sends the transaction to the endorsing peer. On the endorsing peer sides, it decrypts the parameters using the key k in the transient field, and executes the transaction with decrypted parameters. After the transaction execution, the endorsing peers commit the execution result (e.g., hashing it) and put the commitment into the blockchain. Meanwhile, it stores the symmetric key k together with the transaction in its local storage, such that it can provide the key k for accountability check in the future.

It can be observed that the private input of the transaction is not stored in any part of the public known blockchain, instead, the commitments are stored as the proof of the execution. The encrypted transaction is also stored in the blockchain, and the corresponding decryption key is stored in the endorsing peers that execute the transaction. As a result, if the central party (OEM) wants to check the validity of specific transactions, it can ask the endorsing peer to provide relevant symmetric key, decrypt the transaction and execute it, then compare the hash of execution result and the hash stored in the public blockchain. If the hash values match each other, it proves that the endorsing peers executed the transaction honestly. If an endorsing peer refuses to provide corresponding keys, it is itself suspicious and can be considered a malicious party.

[1]As mentioned above, the data in transient field will not be recorded in the blockchain, therefore, the encryption key is only available to the corresponding endorsing peers.

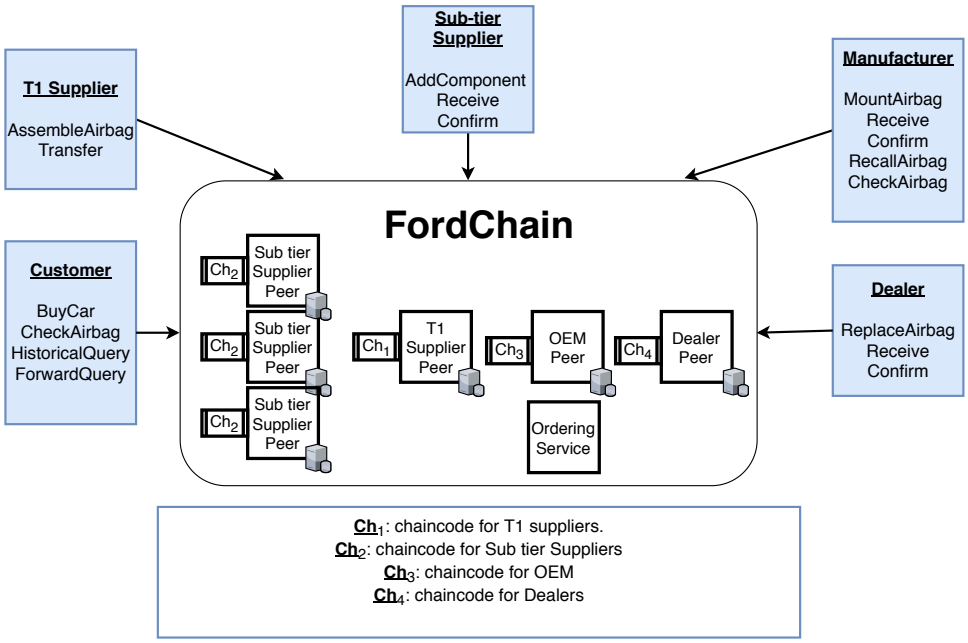

Fig. 5. Chaincode details within FordChain. There are multiple types of endorsing peers such as T1 supplier peers, OEM peers, and dealer peers. They store and execute their owner chaincodes with different functionalities as shown in the figure.

We emphasize that symmetric encryption is not the only solution to provide confidentiality. We pick it in our solution because that it is very efficient and it satisfies all the privacy requirements we need. Other options (e.g., public encryption schemes) can also be applied to achieve the same goal.

4.4.2 Authorization in Fordchain. We make use of multiple chaincodes to ensure authorization. The cornerstone of this approach is that each chaincode can be assigned an endorsement policy and thus different policies can be applied to different chaincodes. Endorsement policies define the smallest set of organizations that are required to endorse a transaction in order for it to be valid. As a result, we can use endorsement policies to enforce business logic (e.g. only tier 1 supplier endorsing peer can endorse assemble_airbag transaction). Due to the different business logic with different requirements, we separate the chaincodes into 4 pieces, each of which has its own endorsement policy: chaincode for Tier 1 supplier, chaincode for sub-tier supplier, chaincode for dealer and chaincode for OEM. When setting up chaincodes, for each peer, we install and instantiate one chaincode and the type of chaincode depends on the peer's identity. Fig. 5 shows an example network with multiple suppliers and their corresponding chaincodes. In such a network, each endorsing peer has its own chaincode with a unique endorsement policy. Additionally, peers are only allowed to endorse transactions that are defined in their chaincodes.

We give more detail in Section 4.6.4 about the challenges we met when separating chaincodes and the corresponding solutions.

4.4.3 Accountability in Fordchain. Achieving accountability is non-trival when privacy protection is designed into the system. Accountability requires that we are able to trace the whole life-cycle of airbags even when data on blockchains are hidden and it should be guaranteed that entities could not lie in their responses.

To begin with, although the data recorded on blockchain is hidden, blockchain could still be used as an unforgeable append-only log. Every transaction is stored in the blockchain and if misbehavior is detected, the entities who endorsed the transaction could be challenged a posteriori for a proof of honest behavior. Although the transaction itself could be in the form of encryption, the encrypted data could be authenticated by signatures so that certain parties will be required to provide the corresponding plaintext. If the party refuses to reveal the data, such fact already implies that the party is malicious. The same logic also applies to the data hidden through commitments.

Below we give an illustrative example of how accountability is preserved in the process of airbag transfer.

The design of Transfer(). The implementation of airbag transfer is divided into two functions: Receive() and Confirm(). Receive() is called by the receiver to claim that a product is sent to the receiver. Similarly, the sender calls Confirm() to claim the receiver successfully received the product. However, with this idea either sender or receiver can just generate thousands of spam transactions claiming some transfer that does not exist at all. Regarding this problem, we design the transfer procedure as follows:

- Sender sends the physical product together with a digital signature to Receiver offline.
- Receiver gets the product, verifies the signature, then sends a Receive() transaction to the blockchain, taking product ID and digital signature as input.
- Receiver sends a confirm message together with a digital signature back to Sender.
- Sender sends a Confirm() transaction to blockchain, taking product ID and Receiver's digital signature as input.

The cornerstone here is the use of the digital signature. Without a valid signature, the transaction itself is invalid and will not pass the chaincode. As a result, both the sender and receiver cannot add spam transactions to blockchain network. Besides, these signatures could be used to prove that endorsing peer provides correct answers for life-cycle query of airbags or components and accountability is achieved.

4.5 Security Analysis

In this subsection we summarize how confidentiality, authorization, and accountability are achieved by our design.

For confidentiality, there are two methods that the adversary could use to access data in Fordchain. One is to eavesdrop on the network traffic and the other is to get data directly from blockchain. In Fordchain, communication among parties is through TLS channels which provides both confidentiality and authentication. As data is available on the blockchain, they are hidden by either commitment or symmetric encryption. As a result, confidentiality is preserved in Fordchain.

Authorization is also achieved in Fordchain. Due to the use of multiple endorsement policies, each endorsing peer can only endorse transactions with their business scope. A transaction is not valid if not enough endorsement signatures are collected. As a result, a party cannot generate a valid transaction beyond the capability of its role.

Accountability is preserved based on two facts. The first observation is that all transactions are recorded in the blockchain, although with some data hidden. The second fact is that the hidden data in blockchain could be opened or revealed in the future. If misbehavior is detected, the corresponding endorsing peers could be asked to open the value hidden in the blockchain. Due to the binding property of crypto primitives that we use, endorsing peers cannot lie about the plaintext of hidden value. When plaintext is revealed it is clear to decide malicious parties.

We put the protection and analysis regarding metadata privacy as a future direction and it is out of the scope of this work.

4.6 System Discussion and Challenges

4.6.1 Multi-tier Supplier Structure. Here we give more details about multi-tier supplier structure. All suppliers are classified to be in different tiers and communication is only allowed between suppliers and suppliers in their neighbor tiers. A sub-tier supplier could provide services to multiple upper layer suppliers, therefore a malicious sub-tier supplier could try to send products with the same id to different upper layer suppliers and Fordchain should be able to detect such malicious behaviors. Besides, privacy protection is significant here since some suppliers make profits through the asymmetry of information. For example, a tier 2 supplier could play as a middleman between tier 1 and tier 3. Such tier 2 just loses its position and profits if tier 1 suppliers get to know tier 3 suppliers directly. As a result, there are various privacy concerns in multi-tier suppliers scenario which are discussed in the rest of the paper.

4.6.2 Car as "Self-aware" Entity. With the technology advancements in the car industry, it is possible now to imagine a future state where we can abstract cars as "self-aware" entities that can verify the mounted components and send signals to invoke or query the blockchain. These advances will bring us a lot of benefits since now a car can check its components occasionally and report faulty components itself. This also builds up the traceability of components in the field.

4.6.3 Correctness of historical query and forward query. The life cycle query is a key feature that industries expect to gain from the combination of blockchain and supply chain. As mentioned above, we leverage digital signatures to guarantee the correctness of the historical/forward query result. Besides, due to the fact that each endorsing peer has its own chaincode and local database, the information regarding life cycle history of airbags is distributed among endorsing peers. For example, an OEM endorsing peer only has knowledge about which tier 1 supplier provides such airbag, but has no idea about information in sub-tier supplier level. This is expected since it is one of the privacy goals for multi-tier supplier structure. As a result, historical query becomes an iterative process where clients send queries to endorsing peers one after another.

4.6.4 Problems due to multiple endorsement policies. As mentioned before, the endorsement policy is used to achieve access control. Therefore chaincode is partitioned into multiple chaincodes, each with their own endorsement policies. In the first version of Fordchain, we design the Transfer() function to be endorsed by both sender and receiver. The idea is straight forward since both sender and receiver should agree on the fact of the transfer. However, this leads to the following problem: Assume there is an airbag which is transferred from party A to B, then from B to C. According to the design there will be two chaincodes, chaincode $Ch1$ endorsed by A and B and chaincode $Ch2$ endorsed by B and C. In the second transfer, the sender should check if the airbag belongs to it or not in $Ch2$, however this information is stored in $Ch1$ since it is a result of first transfer. As a result, $Ch2$ has to invoke $Ch1$ to get this information and unfortunately it is impossible since they have conflict endorsement policies.

To solve this problem we come up with our current solution of transfer which is introduced in Section 4.4.3. There are various advantages to using the current solution. To begin with, all the cases about chaincode invoking another chaincode are eliminated so we do not need to worry about conflicting endorsement policies anymore. Besides, the current solution is more scalable. In the old design, $O(n^2)$ chaincodes are needed to support transfer among n parties. With the current design only $O(n)$ chaincodes are needed.

4.6.5 Considering garage and independent repair shop. In the current model, we do not include garages and independent repair shops. Ideally, following the "right to repair" laws, we can allow the garages and independent repair shops to directly buy legit airbags from the OEM such that they can provide services to the customers. However, a traceability problem can occur when these parties are involved in a replacement process (e.g., they replace the faulty airbags for customers). As these parties do not have endorsing peers, and they have no access to

the blockchain it becomes impossible for them to correctly report the replacement to OEM. Therefore, the OEM loses the traceability of the replacements happening at the repair shops. As the OEM may not know the identity of a replaced component in a car when the car owner get the component added/replaced at the independent repair shops, the OEM may not reach the car owner in case of the component recall.

Nevertheless, we emphasize this only minimally reduces the traceability of the components and our system can be easily updated to tolerate independent repair shops.

4.6.6 Metadata Privacy Issues. Achieving privacy guarantees is non-trivial since there are many ways that information leakage can happen. The blockchain maintains a shared ledger that records plenty of information about the transactions such as the timestamp, chaincode name or the identity of the transaction's creator itself. This information is considered sensitive in many scenarios including supply chain, where different participants wish to hide their business from competitors. In this sense, hiding just the parameter field is far from enough. For instance, the endorser signatures are a critical part within a block, since they indicate which peers checked and endorsed particular transactions. As an illustrative example, assume that only the OEM can mount components into a car and thus only OEM can endorse the relevant transactions. In such a scenario, whenever an adversary observes that a transaction is endorsed by the OEM, the adversary will know that this transaction is calling "MountAirbag" function with a high probability. Similar problems can also appear in relation to the "chaincode name" field in a transaction.

Sensitive data included in the transactions is not only at risk at the chaincode level, but also in the communication between clients and endorsing peers. If the client transmits the transaction in plaintext to the peer, transaction data is trivially leaked to eavesdroppers who can inspect all the traffic. This can be avoided with *authenticated* and *confidential* TLS channels that hide sensitive transaction data from prying eyes of adversaries inspecting the communication between users and endorsing peers. However, although TLS is used, the adversary can still detect the fact that sender and receiver are communicating with each other and this fact itself may reveal information (e.g., transaction frequency and volume implies the business size of a supplier).

In general, using off-the-shelf techniques can assist to build stronger privacy protection at the cost of higher system complexity and reducing its efficiency. Thus, careful analysis and design is required to build upon the appropriate building blocks to maintain the balance between privacy and usability. [17]

5 EVALUATION

We implemented an end-to-end prototype of Fordchain in Hyperledger Fabric version 1.4. The chaincodes are written in Golang. We use the docker images provided by Hyperledger Fabric to build up the networks. Specifically, a docker swarm is used to connect docker networks among multiple physical machines. In the prototype, symmetric encryption and commitments are deployed to achieve confidentiality and accountability.

5.1 Benchmark Settings

To illustrate Fordchain is a practical solution to be deployed in supply chain scenario, we run a series of benchmark to measure the performance of Fordchain in different settings. We deployed our prototypes on two AWS t2.large instances (2 cores and 8GB memory) and the latency between these two machines is less than 10 milliseconds. All peers in Hyperledger Fabric are encapsulated in the form of docker containers so that multiple containers could be run at one physical machine. We distributed the containers in two AWS instances and built up automatic codes so that we can distribute our prototype on multiple AWS instances. Considering hardware is not the bottleneck in our experiment in 2 physical machines setting, we did not increase the number of AWS instances. Docker swarm is used to connect docker containers in multiple physical machines. We run each test case multiple times and took average time as final results. For each test case, we replicate the experiments 10 times and take the average as the final results.

Table 3. Summary of the influence of block size on performance

Block Size	Running Time
5	77.876*s*
10	78.757*s*
20	75.078*s*
50	74.864*s*

To make the evaluation closer to real-world usage, we deploy Raft, a crash fault tolerant consensus protocol, as the ordering service of the prototype. To achieve that five raft containers are bootstrapped to form the raft clusters.

5.2 Evaluation Result

The goal of this benchmark is to confirm that Fordchain is practical to be used in the real world. We confirmed from industry partners that 1000 transactions every day is a proper estimation of the daily load for a supplier. As a result, we tested Fordchain with 1000 *Addcomponent*() transactions. It takes 74.29 seconds for supplier endorsing peers to deal with these transactions if all transactions are valid (e.g., with different component IDs), and 71.07 seconds if these transactions are invalid (e.g., with duplicated IDs). The slight difference may come from the fact that a transaction will be rejected in the middle of chaincode so the chaincode execution time is less. This result shows that Fordchain can easily handle the daily load of suppliers in minutes.

Besides, a proper consensus protocol is required if Fordchain is used in practice. So we compare the performance of solo orderer (i.e., no consensus) and Raft protocol. To achieve Raft consensus, we deployed 5 raft nodes in two AWS instances and repeated 1000 transactions test case. The result is that it takes 78.60 seconds to finish 1000 transactions when Raft is involved. So on average, a 5% latency overhead is added due to the use of Raft, while Fordchain can still deal with the daily load of a supplier easily.

We also tested several parameters that may influence the performance of Hyperledger Fabric such as block size and timeout (Timeout stands for maximum waiting time to form a block). It turns out that these parameters only cause slight changes in latency. For example, in our setting Fordchain gets the best performance when the block size is 50. It takes 74.86 seconds to finish 1000 transactions with Raft consensus. Similarly, block generation timeout also influences the performance slightly, the optimized value varies based on the network connection and transaction speed. Table 3 summarized the performance influence caused by block size.

To conclude, the benchmark successfully shows that our prototype can handle the daily load of supply chain entities efficiently and is practical enough to be deployed for real-world usage.

6 RELATED WORK

The efforts showing the lack of privacy guarantees in Bitcoin [8] bootstrapped a large body of research on privacy-preserving protocols for cryptocurrencies. Initiated by privacy-enhancing overlays for Bitcoin [29, 30] and other payment systems such as Ripple [32], this line of work has resulted on the development of new cryptocurrencies with privacy as the main focus such as Monero [25] and Zerocash [9]. Apart from the privacy aspect, Chod et al. [2] show that decentralized cryptocurrencies can be leveraged to improve transparency in supply chains. These works, however, focus on decentralized cryptocurrencies where the consensus rules are preserved by a dynamic set of unidentified users. This contrasts with the idea of a consortium, where rules are effectively preserved by a set of parties well-known between each other. In the context of permissioned blockchain, there are works using techniques such as ring signatures [21] and trusted hardwares [10] to enhance the privacy of Hyperledger Fabric. Compared to them we propose to use different building blocks and provides different trade-offs.

In the literature, there exist research works that study how to leverage permissioned blockchains for supply chain use cases. Lu et al. [20] provides an overview of the challenges and benefits resulting from the combination of the supply chain and the blockchain. Besides, AlTawy and Gong [6] describe a cryptographic protocol to provide anonymity to the supply chain entities. While this approach is tailored to permissioned blockchains, it does not provide confidentiality of the objects handled in the supply chain. Specifically, more and more industry leaders in automobile are interested in the potential of using blockchain to make mobility safer. For instance, MOBI [1] is a nonprofit smart mobility consortium working with forward thinking companies, governments, and NGOs to make mobility services more efficient and less congested by promoting standards and accelerating the adoption of blockchain, distributed ledger, and related technologies in the mobility industry. There are also several works focusing on the combination of auto supply chains and blockchain such as [23] and [19]. All these works build up the foundation of using Blockchain to secure auto supply chains. Compared with these works, FordChain focuses more on the privacy and ensures that the information about an object traced within the supply chain is only revealed to the object's owners throughout the object's lifetime.

7 CONCLUSION

To conclude, in this work we study the benefits and challenges of using blockchain to prevent counterfeiting in the presence of malicious supply chain parties. In particular, we show that the provision of a distributed and append-only ledger jointly governed by supply-chain parties themselves, by means of a distributed consensus algorithm, makes permissioned blockchains such as Hyperledger Fabric a promising approach towards mitigating counterfeiting. With the combination of blockchain and supply chain and "self-aware" cars, the life cycle traceability of components can be built up even in aftermarket. Authentication, accountability and different level of privacy protection can be also achieved. We also summarized the lessons we learned which can be applied to other supply chain cases and we study the privacy issues in Hyperledger Fabric. Besides, we provide a solution to support access control in Hyperledger Fabric by using multiple chaincodes with multiple endorsement policies.

REFERENCES

[1] Mobility Open Blockchain Initiative. https://dlt.mobi/.

[2] An open-source blockchain protocol for verifiable records, https://dci.mit.edu/b-verify.

[3] IBM and Maersk Struggle to Sign Partners to Shipping Blockchain. https://www.coindesk.com/ibm-blockchain-maersk-shipping-struggling, 2018.

[4] 2013 horse meat scandal, https://en.wikipedia.org/wiki/2013_horse_meat_scandal, 2020.

[5] Nurzhan Zhumabekuly Aitzhan and Davor Svetinovic. Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. *IEEE Transactions on Dependable and Secure Computing*, 15(5):840–852, 2018.

[6] Riham Altawy and Guang Gong. Mesh: A supply chain solution with locally private blockchain transactions. *Proceedings on Privacy Enhancing Technologies*, 2019:149–169, 07 2019.

[7] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, Srinivasan Muralidharan, Chet Murthy, Binh Nguyen, Manish Sethi, Gari Singh, Keith Smith, Alessandro Sorniotti, Chrysoula Stathakopoulou, Marko Vukolić, Sharon Weed Cocco, and Jason Yellick. Hyperledger fabric: A distributed operating system for permissioned blockchains. In *EuroSys*, pages 30:1–30:15, 2018.

[8] Sergey Avdoshin and Alex Lazarenko. Bitcoin users deanonimization methods. *Proceedings of the Institute for System Programming of the RAS*, 30:89–102, 10 2018.

[9] Eli Ben-sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. pages 459–474, 05 2014.

[10] Marcus Brandenburger, Christian Cachin, Rüdiger Kapitza, and Alessandro Sorniotti. Blockchain and trusted computing: Problems, pitfalls, and a solution for hyperledger fabric. *CoRR*, abs/1805.08541, 2018.

[11] Richard Gendal Brown, James Carlyle, Ian Grigg, and Mike Hearn. Corda: An introduction. *R3 CEV, August*, 2016.

[12] Nejdet Delener. International counterfeit marketing: Success without risk. *Review of Business*, 21(1/2):16, 2000.

[13] Daniel Dob. Permissioned vs permissionless blockchains: Understanding the differences, July 2018.

[14] Ali Dorri, Salil S Kanhere, Raja Jurdak, and Praveen Gauravaram. Blockchain for iot security and privacy: The case study of a smart home. In *Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference*, pages 618–623,

2017.
[15] Sheila Frankel, K. Robert Glenn, and Scott G. Kelly. The AES-CBC Cipher Algorithm and Its Use with IPsec. RFC 3602, September 2003.
[16] David Galvin. IBM and Walmart: Blockchain for Food Safety. https://blockchain578369954.files.wordpress.com/2018/06/6-using-blockchain-for-food-safe-2.pdf, 2017.
[17] Ryan Henry, Amir Herzberg, and Aniket Kate. Blockchain Access Privacy: Challenges and Directions. *IEEE Security & Privacy*, 16(4):38–45, 2018.
[18] Don Johnson, Alfred Menezes, and Scott Vanstone. The elliptic curve digital signature algorithm (ecdsa). 1(1):36âĂŞ63, August 2001.
[19] Tim Reimers ; Felix Leber ; Ulrike Lechner. Integration of blockchain and internet of things in a car supply chain. 04 2019.
[20] D. Lu, P. Moreno-Sanchez, A. Zeryihun, S. Bajpayi, S. Yin, K. Feldman, J. Kosofsky, P. Mitra, and A. Kate. Reducing automotive counterfeiting using blockchain: Benefits and challenges. In *2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON)*, pages 39–48, 2019.
[21] Subhra Mazumdar and Sushmita Ruj. Design of anonymous endorsement system in hyperledger fabric. *CoRR*, abs/1811.01410, 2018.
[22] Carlos Mena, Andrew Humphries, and Thomas Y Choi. Toward a theory of multi-tier supply chain management. *Journal of Supply Chain Management*, 49(2):58–77, 2013.
[23] Daniel Miehle, Dominic Henze, Andreas Seitz, Andre Luckow, and Bernd Bruegge. Partchain: A decentralized traceability application for multi-tier supply chain networks in the automotive industry. pages 140–145, 04 2019.
[24] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.
[25] Shen Noether. Ring signature confidential transactions for monero. *IACR Cryptology ePrint Archive*, 2015:1098, 2015.
[26] U.S. Department of Commerce, National Institute of Standards, and Technology. *Secure Hash Standard - SHS: Federal Information Processing Standards Publication 180-4*. CreateSpace Independent Publishing Platform, North Charleston, SC, USA, 2012.
[27] Marc Pilkington. Blockchain technology: principles and applications. *Research handbook on digital transformations*, page 225, 2016.
[28] Eric Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446, August 2018.
[29] Tim Ruffing, Pedro Moreno-Sanchez, and Aniket Kate. Coinshuffle: Practical decentralized coin mixing for bitcoin. In Mirosław Kutyłowski and Jaideep Vaidya, editors, *Computer Security - ESORICS 2014*, 2014.
[30] Tim Ruffing, Pedro Moreno-Sanchez, and Aniket Kate. P2p mixing and unlinkable bitcoin transactions. 01 2017.
[31] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy (S&P)*, pages 459–474.
[32] David Schwartz, Noah Youngs, Arthur Britto, et al. The ripple protocol consensus algorithm. *Ripple Labs Inc White Paper*, 5:8, 2014.
[33] G. E. Suh and S. Devadas. Physical unclonable functions for device authentication and secret key generation. In *2007 44th ACM/IEEE Design Automation Conference*, pages 9–14, June 2007.
[34] Feng Tian. An agri-food supply chain traceability system for china based on rfid & blockchain technology. In *Service Systems and Service Management (ICSSSM)*, pages 1–6, 2016.
[35] Sarah Underwood. Blockchain beyond bitcoin. *Communications of the ACM*, (11):15–17, 2016.
[36] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 2014.
[37] Xiao Yue, Huiju Wang, Dawei Jin, Mingqiang Li, and Wei Jiang. Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. *Journal of medical systems*, 40(10):218, 2016.
[38] Guy Zyskind, Oz Nathan, et al. Decentralizing privacy: Using blockchain to protect personal data. In *Security and Privacy Workshops (SPW), 2015 IEEE*, pages 180–184, 2015.