IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier

A Blockchain-Based Framework for Supply Chain Provenance

PINCHEN CUI[1], (Student Member, IEEE), JULIE DIXON[1], UJJWAL GUIN[2],(MEMBER, IEEE), AND DANIEL DIMASE[3], (Member, IEEE).

[1]Deparment of Computer Science and Software Engineering, Auburn University, AL 36849 USA (e-mail: Pinchen@auburn.edu)
[2]Department of Electrical and Computer Engineering, Auburn University, AL 36849 USA (e-mail: ujjwal.guin@auburn.edu)
[3]Aerocyonics, RI 02818, USA (e-mail: daniel.dimase@aerocyonics.com)

Corresponding author: Pinchen Cui (e-mail: Pinchen@auburn.edu).

This work was supported in part by the National Science Foundation under Grant Number CNS-1755733 and Intramural Grants Program (IGP) from Auburn University.

ABSTRACT The complexity of the electronics supply chain has grown significantly due to the expansion of globalization in the 21st century. Electronic parts are now manufactured, distributed, and sold globally. Ensuring the security and integrity of the supply chain has become extremely challenging due to the widespread infiltration of untrusted hardware, specifically, counterfeit and cloned parts. Especially, the provenance of microelectronics and commercial off-the-shelf (COTS) parts becomes prohibitively difficult to track and calls for immediate solutions. In this paper, we present a non-destructive way of ensuring the traceability of electronic parts in the supply chain. We have implemented a blockchain-based framework, which helps to track and trace every chip while they are circulating in the supply chain. The proposed framework is built upon a permissioned blockchain. Hyperledger is used for implementing this framework. A detailed analysis is carried out to present the feasibility of our proposed approach.

INDEX TERMS Internet of Things (IoT), Cyber-Physical Systems (CPS), Physically Unclonable Functions (PUF), Edge Device, Cloning, Blockchains, Device Identity, Track and Trace

I. INTRODUCTION

Due to the rise of globalization, it is now extremely challenging to ensure the security and integrity of the electronics supply chain. Numerous reports pointed out the widespread infiltration of counterfeit integrated circuits (ICs) in our critical infrastructures. The majority of these reports highlights ICs that are reclaimed from the used and discarded electronic waste, and are commonly known as recycled ICs [1]. Information Handling Services reported that the potential annual risk of the global supply chain from counterfeiting is at $169 billion and increasing [2]. As the operational life of our critical infrastructures (*e.g.,* various defense and aerospace systems) are much longer than the life of electronic parts, it is necessary to obtain obsolete parts, which are no longer in production by the original component manufacturers (OCMs), from untrusted third party suppliers who are often located offshore [3], [4]. In addition, cloned parts are also on the rise [5]–[7]. Recently, the groundbreaking hardware hack on the supply chain, introduced by Bloomberg in October 2018, actually sets off an alarm [8]. The article reported an example of a state sponsored vulnerability accomplished through the insertion of a tiny microchip, not much bigger than a grain of rice, that wasn't part of the boards' original design. According to the article, investigators determined that the chips allowed the attackers to create a stealth doorway into any network that included the altered machines. By compromising the supply chain, adversaries could effect well-known top United States companies and government services.

Aiming to address and respond to the supply chain security problems, the U.S. Department of Defense (DoD) has introduced a new supply chain risk management strategy named "Deliver Uncompromised" [9] which aims to secure and ensure the deliveries of military and government supply chains. In addition, the National Institute of Standards and Technology (NIST) also updated their cyber security framework with new supply chain security definitions and policies [10] where a supply chain management category has been added into the framework core. However, while some criteria and policies are established, the real world implementation and practice are still in infancy and evolving.

A. CONTRIBUTIONS

The detection of a compromised device is extremely challenging as there are a wide variety of parts with different resources already in the supply chain. Finding a *one-size-fits-all* solution is our primary objective such that the majority of devices can be authenticated using this single solution. Ensuring the security of the supply chain requires the authenticity for all parts, which can be guaranteed if we can track the parts through trusted suppliers back to their true origin. To an extent, some level of protection exists today that addresses the detection of counterfeit and cloned devices, however, a complete solution for the traceabilty of a part in the supply chain is yet to be developed. In this paper, we propose to use blockchain technology to ensure the security and integrity of the supply chain by enabling traceability of electronic parts. In our design, blockchain and smart contracts enable the reliable traceability and verification for parts, while they travel in the supply chain. A “smart contract” is a computer protocol intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract. Smart contracts allow the performance of credible transactions without third parties. These transactions are trackable and irreversible. In addition, a smart contract is used more specifically in the general purpose computation that takes place on a blockchain using cryptographic hash chains. The major contributions of this paper are summarized as follows:

- We proposed a novel architecture, which uses a low-cost blockchain instance for providing traceability of electronic parts or devices. The traceability is ensured using a unique device ID, which can be programmed into the device using one-time programmable memory (commonly known as electronic chip ID or ECID [11]), or a unique identification can be obtained for a physically unclonable function (PUF) [12]–[16]. The detection of counterfeit ICs (primarily the recycled ones) can be ensured using ECID, while PUFs can provide protection against cloning. The origin of a device, the trace of travel in the supply chain, and its bill of materials can be accurately tracked, and this solution can used for verifying its authenticity. A simple query to the blockchain will show all the necessary information for such a purpose.
- To the best of our knowledge, our proposed blockchain-based framework is the first approach that comprehensively addresses in-transit thefts, human errors, delivery and management failures, and dishonest entities in the supply chain in a comprehensive way. Note that a device ownership transfer generally can be triggered and controlled by device owners in any traditional blockchain-based solutions. Wrong electronic parts could accidentally be sent, which leads to the inappropriate ownership transfer. Logistic and transportation could be delayed or failed due to external causes (*e.g.,* weather, natural disasters, etc.), the receiver of a part may cancel the original order even when the ownership of a part have already been transferred in the blockchain. Parts could also be stolen by adversaries during the transportation. Note that these stolen parts are still valid since ownerships are already transferred to a trusted participant in the blockchain. Moreover, a receiver of parts can deny the transfer or acceptance of part after the ownership transfer is completed. Therefore, directly transferring the ownership within one transaction creates irreversible results in the traditional blockchain-based systems, which could cause further security and management risks. To address these aforementioned threats, we propose a confirmation-based ownership transfer in our blockchain-based framework for enabling device traceability. In this proposed framework, a two-transaction-based ownership management is proposed. After the ownership transfer transaction has been sent by the sender, an additional confirmation transaction from the receiver is required. The ownership transfer will be completed once the mutual agreement between sender and receiver is reached. This will automatically tag the items, which are missing during the transportation, human errors, and delivery failures.
- We implemented a prototype system to demonstrate the feasibility of our proposed approach. A permissioned blockchain (Hyperledger Fabric [17]) is used along with a non-resource intensive consensus algorithm, where most of the previous works were implemented via Proof of Work (PoW) based permissionless blockchain (*e.g.,* Ethereum). The features of consortium blockchain and Hyperledger eliminate the cost of a transaction fee and improve the efficiency by using a non-resource intensive consensus algorithm.

B. RELATED WORK

A significant amount of research has been directed to ensure the security and integrity of the supply chain by the efficient detection and avoidance of counterfeit ICs [1], [18]–[30]. The approaches can be categorized into different categories – (i) standards [18], [31]–[33], (ii) statistical data analysis [22]–[25], [34], [35], (iii) on-chip sensors and structures [27]–[29], [36]–[39] and (iv) unique markers [40]. Even though these solutions can provide some levels of detection of counterfeit ICs, none of them can provide the traceability information, such as the origin, manufacturer, bill of materials, and travel trace in the supply chain.

The integration of blockchain and supply chain receive widespread attention, since the inherent properties and features of blockchain could significantly enhance the traceability, transparency, and reliability of the supply chain [41], [42]. Some researchers discussed, proposed, and analyzed various blockchain based frameworks to refine the traceability for supply chain [43]–[52]. By leveraging the blockchain, the traceability of food [44], [47], [51], healthcare [48], [52] and post delivery supply chain [43] could be enhanced. Contrary to the traditional blockchain-based tracking (*e.g.,* food and healthcare products), electronic devices possess an advantage of integrating unclonable ID, which can be generated from a PUF embedded into the device, and thus can enable efficient and low-cost tracking (*e.g.,* registration, verification, and status update).

The authors in [53] introduced a blockchain-based framework, which ensures the authenticity of electronics with the help of an unclonable ID generated from a SRAM-based PUF. Xu *et al.* provided a comprehensive solution and summary for using blockchain to improve and secure the integrity of electronic supply chain [54]. However, these two solutions do not provide detailed traceability and ownership information for a device. Islam *et al.* proposed a method that uses PUF and blockchain to enhance authenticity and traceability of parts in the supply chain [55]. However, the device ownership transfer is simply triggered and controlled by device owners. This design may lead to potential security issues. Human errors, delivery and management failures, in-transit thefts, and dishonest participants are still threatening supply chain even with implementation of blockchain for tracking [56].

Note that blockchain was first introduced by Bitcoin [57] and is now widely used by the cryptocurrencies. Blockchain is known as a distributed and shared digital ledger, where all the transactions and records are hashed and stored in the chain to provide both integrity and transparency. Certain blockchains also support the smart contract [17], [58] which allows the user to run Turing-complete scripts on the chain. Using a smart contract (also known as chaincode in Hyperledger) enables the user to store and manage data inside of the blockchain, various of applications such as Filecoins [59] and Storj [60] have been proposed.

C. ORGANIZATION

The rest of the paper is organized as follows: we introduce our proposed novel blockchain-based framework in Section II. The implementation details are described in Section III. The analysis of our design are performed in Section IV. Finally, we conclude our paper in Section V.

II. PROPOSED BLOCKCHAIN-BASED FRAMEWORK FOR SUPPLY CHAIN PROVENANCE

Ensuring traceability for devices is critical for providing trust among different entities in the electronics supply chain. This section presents a blockchain-based framework to allow an entity to track electronic devices. Figure 1 describes a simplified version of the supply chain, which consists of five different types of entities – design authority, contract manufacturer, distributor, end user/customer, and adversary. The raw material and logistics service providers are omitted in this model for simplicity. Even our simplified model demonstrates the complexities of the supply chain with a limitless number of possibilities for an adversary to introduce their compromised product. Note that, a design authority (DA) can be described as entity in the supply chain who owns the intellectual property (IP) of a design and could produce the device or assembly or have their product produced by a contract manufacturer. Many of the DAs of microelectronic devices do not own a manufacturing plant (foundry or fab) and outsource the fabrication to contract manufacturers due to the prohibitively high-cost of building and maintaining a foundry [61]. Once the chips are fabricated from a foundry, two possible distribution scenarios may occur – (i) the design authority could ask the contract manufacturer to send back all the parts, and distribute them by itself, or (ii) the contract manufacturer directly sends the parts to the customer or DA authorized distributors. Note that, many of the distributors in the supply chain may not be authorized by the design authority to distribute their parts. Distributors that are not authorized by the design authority are often called Independent Distributors or Brokers.

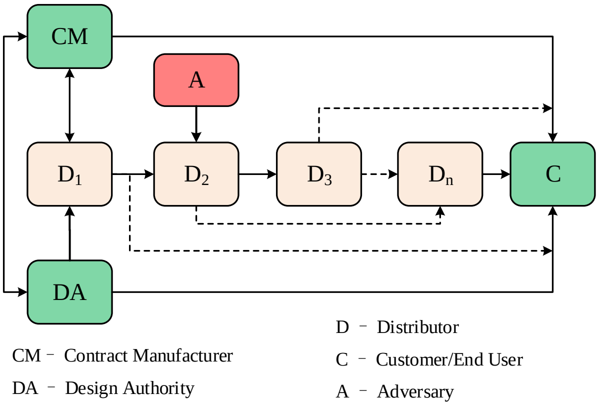


Figure 1: An abstract view of the electronics supply chain consisting of a design authority, a contract manufacturer, several distributors, a customer and two adversaries.

Figure 1 shows an abstract view of the electronics supply chain that consists of a design authority (DA), a contract manufacturer (CM), several distributors (D_{1-n}), an end user/customer (C), and an adversary (A). We generally treat the design authority, contract manufacturer, and the customer as trusted and highlighted in green. The distributors can be of both (trusted and untrusted) types and highlighted in light brown, whereas the adversaries are always untrusted and highlighted in red. The adversary A can make cloned devices, or can integrate counterfeit (recycled) devices or tampered devices with hardware Trojans or malware into the supply chain. To address this problem, it is necessary for the customer C to track the origin and the trace of the devices travelled in the supply chain. It is absolutely necessary to develop a framework that can provide the traceability, in which the trace of legitimate devices ($CM - C$ or $DA - C$ or $DA - D_1 - C$ or $CM - D_1 - D_2 - D_3 - C$ or $CM - D_1 - D_2 - D_n - C$) could be verified effortlessly.

We propose to use a blockchain-based architecture to provide a comprehensive, persistent and reliable device tracking and verification service for different manufacturers, distributors and customers. By using blockchain, all the entities will be able to securely record the device ownership trans-

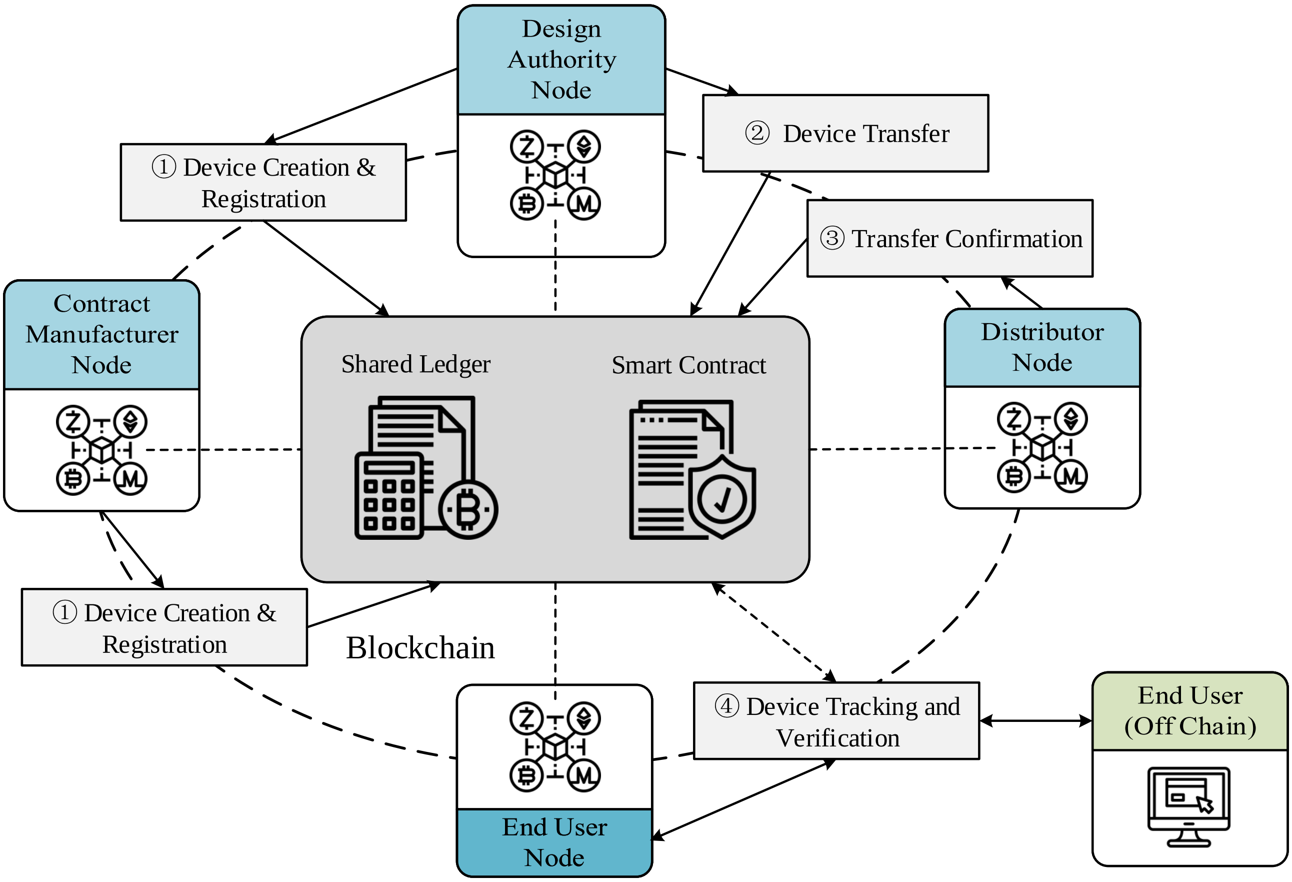


Figure 2: Proposed Blockchain for Supply Chain Provenance.

fer, meanwhile tracking and then verifying the authenticity of each device. The design of the architecture is shown in Figure 2. The framework is built upon a consortium-based blockchain, which consists of four types of nodes: design authority, contract manufacturer, distributor, and end user/customer. The overall design is demonstrated in Section II-A. Using smart contracts, four operations are designed to enable the traceability. The design authority and contract manufacturer could register the devices into the blockchain by using a device registration function, which is introduced in Section II-B. The transfer of the devices in the blockchain could be performed by using a device transfer transaction, as illustrated in Section II-C. In order to provide additional security, the new owner of the devices needs to send a transfer confirmation to complete the ownership transfer. This procedure is described in Section II-D. Finally, the end user could track the trace of the devices and verify the authenticity by using the device tracking and verification function, which is introduced in Section II-E.

A. CONSORTIUM-BASED BLOCKCHAIN

Generally, consortium blockchain is a permissoned blockchain formed by a group of known and verified members. The consortium blockchain (*e.g.,* Hyperledger) eliminates the cost of a transaction fee and improves the efficiency by using a non-resource intensive consensus algorithm. As a result, a consortium blockchain could minimize the cost of the daily operations in the supply chain, which is ideal for building a supply chain tracking system.

In this blockchain based system, design authority, contract manufacturers, and distributors are the major members of blockchain and they have to be registered as "nodes" in blockchain. Each of the nodes must create and maintain an identity (i.e., address, account or a participant identity) in the system. Any addition (new member), replacement or duplication of identities must be notified to and accepted by all the major members identified in the chain. The major members could be notified based on a buyer/seller transaction or a more broad based notification system based on the security needs of the transaction. A customer could also be registered with an identity in the blockchain. In such way, the post-sale traces could be recorded in the blockchain as well. If the customer is not registered in the blockchain infrastructure, any re-distribution of the devices confronts risks the security since the re-distribution procedure is not officially certificated and protected by the provenance system.

On the other hand, the underlying functionalities that provide the actual data storage and management are implemented by smart contract or chaincode. The smart contract or the chaincode needs to be internally advertised and distributed, and all the entities have to install the scripts locally. The creation, maintenance and deprecation of the scripts need to be verified by all the major members identified in the chain. This procedure could be performed on-chain or off-chain. One blockchain could run multiple smart contracts to maintain and manage different types of devices.

B. INITIALIZATION AND REGISTRATION OF DEVICE

The design authority and contract manufacturers could register devices type (i.e., electronic device types) on the chain. Each type of device needs to be registered separately in the blockchain. This could be achieved by using smart contracts. The creation and registration of these device types are written into the blockchain and are visible to all the members downstream in the chain and by those major members identified upstream who require notification. Note that, only the design authority and contract manufacturers could register the device types in the blockchain system. Any other participants like distributors and customers could not register the device type in the system due to the defined blockchain policy.

Once the device types are registered in the blockchain, the design authority or contract manufacturer needs to register the devices manufactured from the production line. For traceability purposes, a unique device ID is necessary, which can be easily constructed by integrating an ECID, PUF or a unique identification to the device. Instead of placing the ID directly to blockchain, we propose to store the hash of the ID. This provides an additional security as it prevents one to determine the original ID unless he/she actually possesses the device. For each device, design authority or contract manufacturer needs to upload the hash of the ID into the blockchain (*e.g.,* stored in an array in the smart contract). All the uploaded hashes and the number of the entries are known by all the members, however, since the IDs are hashed, none of the actual IDs would be leaked. In addition, the same logic applies with the device type registration; only the design authority and contract manufacturer producing the devices are allowed to register the devices into the blockchain.

Let us now consider an example, suppose contract manufacturer (CM) registers device type H in the blockchain, it can then upload the same type of devices produced from a manufacturing unit. If it wants to register N devices, it need to compute N hashed IDs and upload into the chain under device type H. Note that it cannot upload any device ID except for this device type. However, it can register another device type under class K and upload K-type devices into the chain. This procedure is depicted in the Figure 3. The registration procedure relies on the implementation of smart contracts, and all the data of the contracts are stored in the blockchain ledger.

C. TRANSACTION FOR DEVICE TRANSFER

To ensure traceability, it is required to record the transfer of an device among different entities in the supply chain. This can easily be implemented in our proposed blockchain-based framework shown in Figure 3. Initially, there are N copies of device type H stored in the blockchain. If the contract manufacturer CM decides to send N_1 copies of device H to Distributor D_1, a specific device transfer transaction needs to be sent, including the data of the devices. In addition, a smart contract/chaincode would be triggered by this transfer transaction to perform further processing (change the data stored in the blockchain). Besides the normal fields of the transaction (details varies on different blockchain platforms), the transaction for a device transfer may need four additional elements in the payload: the device type that is being transferred, the amount of a device, the identifiers (IDs) of the devices, and the new owner of the device. As shown in the Figure 3, contract manufacturer CM sends a device transfer transaction with additional data (H, N_1, ID_1, D_1) which represent N_1 of H with device ID_1 would be transferred to Distributor D_1 $(N_1 \leq N)$. Note that, for transferring N_1 entries in the blockchain, depending on the implementation, the transaction could be N_1 transactions, and each of them contains one hashed ID, or only one transaction (or several) contains all the hashed IDs.

Generally, design authority, contract manufacturers, and distributors are allowed to initiate transactions for the device transfer, as long as they own a certain amount of devices. However, the actual ownership of the device that is declared to be transferred in the device transfer transaction would not be transferred to the new owner until a confirmation transaction is received. The primary reasons for receiving an additional confirmation transaction from the receiver of the device (devices) are the following. First, the receiver of an device must acknowledge the number of received items, such that every device is accounted for. The receiver needs to compare the hash of the device IDs with the hash stored in the chain. If any mismatch is found, the transaction will be cancelled due to this compromised device. Appropriate parties in the chain will be notified to take appropriate actions when there are mismatches. Second, the receiver cannot deny the acceptance of the delivered shipment. Without the confirmation, none of the devices in the shipment is recorded as legitimate transferred in the blockchain. Finally, one can easily track missing devices that never reach the receiver. This could be helpful if an adversary intercepts the shipment and stole devices during the transit.

D. TRANSACTION FOR DEVICE TRANSFER CONFIRMATION

Upon receiving a specific number of electronic devices from an entity (e.g., a manufacturer or a distributor), the new owner of the device needs to send out the confirmation transaction. A device transfer is not completed and verified until a confirmation of the transfer has been made. The trace and the ownership of the device would be transferred in the smart contract only after the confirmation.

Figure 3 shows the detailed process of how a transaction is first created, then validated, and finally added to the blockchain. At step 1, device registration is completed. Contract manufacturer CM registers device type H to the

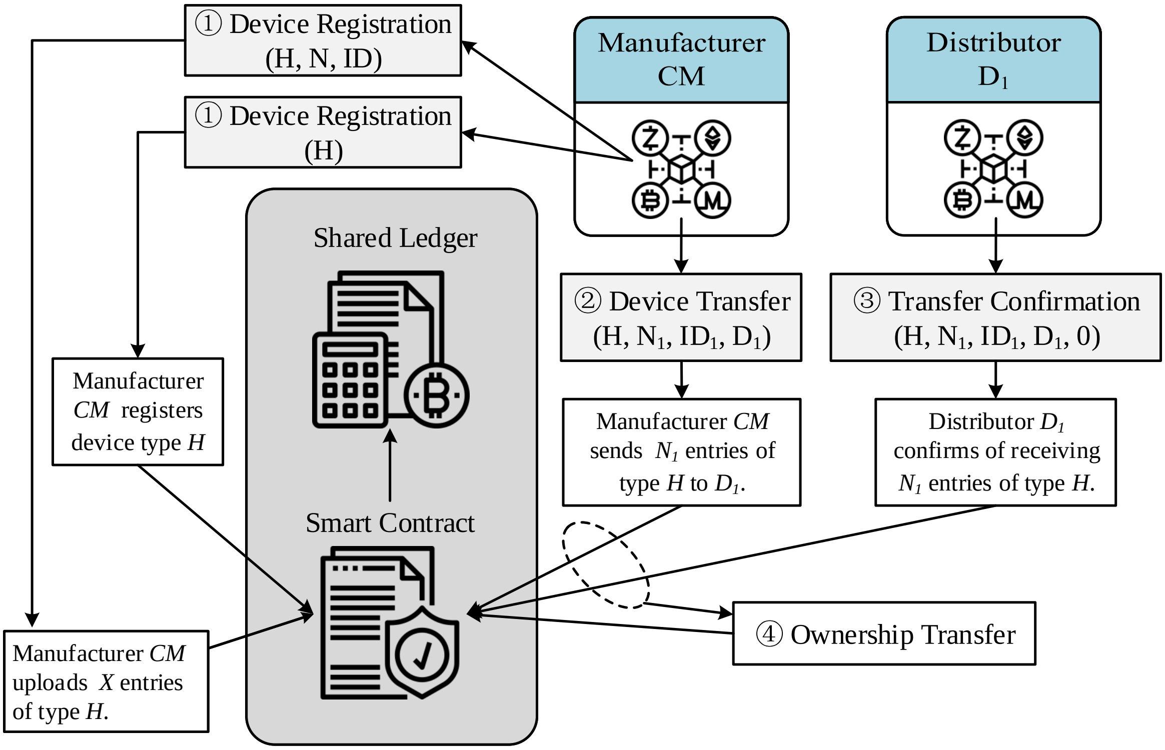

① Device Registration (H, N, ID)
① Device Registration (H)
Manufacturer CM
Distributor D_1
Shared Ledger
② Device Transfer (H, N_1, ID_1, D_1)
③ Transfer Confirmation (H, N_1, ID_1, D_1, 0)
Manufacturer *CM* registers device type *H*
Manufacturer *CM* sends N_1 entries of type *H* to D_1.
Distributor D_1 confirms of receiving N_1 entries of type *H*.
Smart Contract
④ Ownership Transfer
Manufacturer *CM* uploads *X* entries of type *H*.

Figure 3: Operations and Transactions in the Proposed Blockchain-based Framework.

blockchain. It is now able to perform transactions. At step 2, the CM initiates a transaction, which contains the transfer of N_1 number of H device to the distributor D_1. After physically receiving N_1 numbers of device H from CM, D_1 needs to verify all the device IDs with the stored hashed IDs in the blockchain (the details of verification is demonstrated in Section II-E). Note that a failed verification not only invalidates the previous transaction, but also notifies appropriate parties in the chain to address concerns. At step 3, the distributor D_1 initiates the device transfer confirmation transaction once the verification is complete. The distributor D_1 sends out the confirmation transaction with elements $(H, N_1, ID_1, D_1, 0)$. The last field in this transaction payload represents the status of the confirmation process. One can assign 0 for successful transaction. At step 4, after the confirmation transaction is received, the smart contract (chaincode) would then transfer the ownership of the N_1 of H to Distributor D_1. It is reasonable to consider some special cases, for instance, a transaction is failed due to the mismatched items that were sent, or a part of the items are mismatched. One could assign other values to the last field of the transaction payload to indicate additional status regarding the transaction (e.g. partial shipment, damaged, etc.).

Note that, in this system, the smart contract keeps track of the unconfirmed device transfer transaction. Only the valid receivers (that have unconfirmed transfers) could initiate the confirmation transactions. In addition, the *time-to-live* of the device transfer could be enabled to define the expiration time of the transfer. When a transfer is accidentally created, or the receiver node is failed, the transfer transaction could be set to automatically expire after a certain period of time.

E. VERIFICATION AND TRACKING

Whenever a participant physically receives a device, it is required to verify its identity (ID) which is present (hashed) in the blockchain. The verification procedure requires the retrieval of the unique device ID, which can be accessed by using JTAG interface [62] or other unique identification methods that are tamper proof. One could verify the hashed device ID with the hashed ID records stored in the blockchain through the blockchain query functions (details are described in Section III-B). The original manufacturer, current owner, and other major members identified for the chain could be alerted with information that includes historical traces of the device as a result of this query. If the ID does not exist in the system, a flag will be raised and the device will be identified as suspicious. Note that, the verification and tracking procedure do not alter the data stored in the blockchain, thus no actual transaction would be made and the entire procedure is highly efficient.

F. MINING IN PROPOSED PROVENANCE SYSTEM

Regardless which consensus algorithm would be applied in the blockchain-based framework, one of the most crucial configuration set ups is to decide the miners. Mining, in the context of blockchain technology, is the process of adding transactions to the ledger of existing transactions, known as the blockchain. According to the design and the purpose of the system, all the major members (manufacturers and distributors) of the blockchain are permissioned and known. It is reasonable and reliable to adopt all the major members to be valid and potential miners (endorsers).

III. IMPLEMENTATION DETAILS

As Hyperledger Fabric is becoming one of the most promising and successful blockchain platforms, it is our selection of framework. Hyperledger Fabric is introduced and maintained by IBM [17]. The native permissioned architecture and non-resource intensive consensus mechanism of Hyperledger perfectly matches the requirements of implementing the proposed blockchain.

A. BLOCKCHAIN NETWORK MODEL

The Hyperledger blockchain network model which includes the underlying data structures used in our proposed implementation is shown in Figure 4. The type *entity* defines the blockchain participants with corresponding attribute (*e.g.,* design authority, contract manufacturer, distributor, and customer). An type *deviceTypes* is created to define the types of devices, which needs to be and can only be registered by the design authority and contract manufacturer. These device types represent different types of devices already circulating in the supply chain. For example, Intel Pentium processor can be a device type. In order to avoid a device to be sent multiple times before its confirmation has been made, an enum *transferStatus* declares the transfer status of the device (*e.g., NOT_IN_TRANSFER*, *IN_TRANSFER*). This status attribute helps to identify whether a device is available for transfer. The type *device* consists of following attributes: device ID, device type, transfer status, receiver of the transfer (if exist), original manufacturer, current owner and device traces. Finally, four functions *register*, *transfer*, *confirmation* and *query* are defined. The detailed transaction processing functions are introduced in the following section (Section III-B). Note that, the type *entity* should be maintained by the blockchain admin in the chaincode, so that the participants are regulated permissioned to access the chaincode.

```
type entity{};
type deviceTypes{};
enum transferStatus{};
type device{};
function register();
function transfer();
function confirmation();
function query();
```

Figure 4: Data Structure of the Blockchain Network Model

B. CHAINCODE IMPLEMENTATION

The registration of devices and device types are carried out by the *register* function, which is described in Algorithm 1. The *register* function requires three arguments: a registration flag, a device type, and a device ID. First, the function checks that whether the caller is valid DA or CM, if the check fails, an error message is returned. If the flag is set to 0, it means the call is for device type registration. The function would then create a device type and verify that it has not been registered in chain, and finally store it into blockchain. On the other hand, if flag is set to 1, this function would create and initialize a device data record in blockchain with provided device type and hashed device ID. If the flag is set to some other values, the function returns error.

Algorithm 1: Pseudocode for register function ***Register()***

```
Input  : Registration Flag (F), Device Type (T), Hashed
         Device ID (I)
if CurrentParticipant is not a DA or MC then
    throw error message
end
if $F = 0$ then
    thisType = deviceType thisType;
    thisError = Fetch type record with key T;
    if thisError != null then
        throw type is registered error
    end
    else
        Update (thisType) in Chaincode;
    end
end
else if $F = 1$ then
    thisDevice = device thisDevice;
    thisDevice.deviceType = T;
    thisDevice.deviceID = I;
    thisDevice.transferStatus = NOT_IN_TRANSFER;
    thisDevice.owner = CurrentParticipant;
    thisDevice.trace = [CurrentParticipant];
    Update thisDevice in Chaincode;
end
else
    throw error message
end
```

Algorithm 2: Pseudocode for device transfer function ***deviceTransfer()***

```
Input  : Hashed Device ID (I), Receiver Entity (R)
device = Fetch device record with key I;
if device.owner != CurrentParticipant then
    throw error message
end
if device.transferStatus != NOT_IN_TRANSFER then
    throw error message
end
device.transferStatus = IN_TRANSFER;
device.transferTo = R;
Update device in Chaincode;
```

Algorithm 2 illustrates *deviceTransfer()* function, which is used to transfer devices. The function first retrieves the device data record from blockchain. If the current transfer transaction sender is not the owner of the device, the function fails. Then the function checks whether the device is available for transferring, and then updates the status and data. Note that, the transfer function handler only updates the basic information of the device. The actual ownership and the trace of the device would not be updated. This pseudocode only describes the scenario of transferring a single device, and one could easily alter it to transfer certain amount of a particular type of device. In addition, the expiration time could be enabled as an optional feature (introduced in Section II-D), which is omitted in the algorithm.

Algorithm 3: Pseudo-Code of device Transfer Confirmation Function ***TransferConfirmation()***

```
Input   : Hashed Device ID (I)
device = Fetch device record with key I;
if device.transferStatus != IN_TRANSFER then
    throw error message
end
if device.transferTo != CurrentParticipant then
    throw error message
end
device.transferStatus = NOT_IN_TRANSFER;
device.owner= CurrentParticipant;
device.trace.append(CurrentParticipant);
Update device in Chaincode;
```

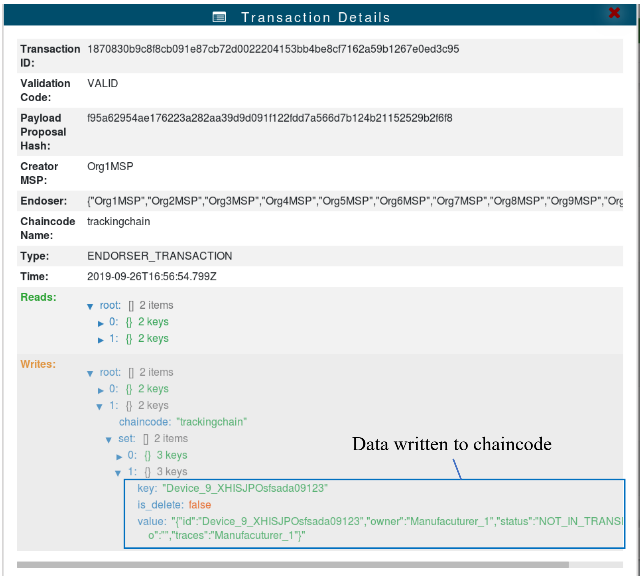
The Algorithm 3 describes Transfer Confirmation Function *TransferConfirmation()*. Given a successful confirmation, the actual ownership of the device would be transferred and the trace of the device would be updated. The function first needs to check the validity of the transaction creator, and then update the data of the device. Note that the failed, partial, and complete transfer confirmation flag (mentioned in Section II-D) could be enabled as an option.

The Algorithm 4 describes the details of tracking and verification function. This function is performed by using the query feature of the Hyperledger system. There are two types of queries: first type is the normal query, as shown in Algorithm 4, which simply returns the data stored in the chaincode with mapped keyword. Another type of query is rich query, which could deeply utilize the underlying mechanism of state database and enables the user to perform SQL-like queries. For instance, one could send a SQL-like query to retrieve all the devices belonging to one distributor. In addition, the query functions could be exposed to the users via API/Webpage (RESTful API [63]). One could trigger the query with the hashed device ID to get the corresponding device information from outside of the infrastructure.

Algorithm 4: Pseudocode for Tracking and Verification Function, ***TrackAndVerify()***

```
Input: Hash of the device IDs, $HIDs$
Fetch the device data record with key (HID);
```

Note that, *query* transactions will not be appended into blockchain, since it only queries the data stored in chaincode without altering it. On the other hand, all the transactions invoked by *register*, *transfer*, *confirmation* functions will be appended into blockchain. A sample transaction details is shown in Figure 5. All the write operations and data stored in blockchain are recorded in transaction history with details. Later *query* functions will initiate transactions to retrieve the stored data from the chaincode. Moreover, one could also use blockchain explorers (*i.e.,* Hyperledger Explorer [64]) to manually check the transaction details as well.



Transaction Details

Transaction ID:	1870830b9c8f8cb091e87cb72d0022204153bb4be8cf7162a59b1267e0ed3c95
Validation Code:	VALID
Payload Proposal Hash:	f95a62954ae176223a282aa39d9d091f122fdd7a566d7b124b21152529b2f6f8
Creator MSP:	Org1MSP
Endoser:	{"Org1MSP","Org2MSP","Org3MSP","Org4MSP","Org5MSP","Org6MSP","Org7MSP","Org8MSP","Org9MSP","Org
Chaincode Name:	trackingchain
Type:	ENDORSER_TRANSACTION
Time:	2019-09-26T16:56:54.799Z

```
Reads:
    ▼ root: [] 2 items
      ▶ 0: {} 2 keys
      ▶ 1: {} 2 keys

Writes:
    ▼ root: [] 2 items
      ▶ 0: {} 2 keys
      ▼ 1: {} 2 keys
          chaincode: "trackingchain"
        ▼ set: [] 2 items
          ▶ 0: {} 3 keys
          ▼ 1: {} 3 keys
              key: "Device_9_XHISJPOsfsada09123"
              is_delete: false
              value: "{"id":"Device_9_XHISJPOsfsada09123","owner":"Manufacuturer_1","status":"NOT_IN_TRANSI
                o":"","traces":"Manufacuturer_1"}"
```

Data written to chaincode

Figure 5: A snapshot of the transaction details in Hyperledger Fabric blockchain.

C. ACCESS CONTROL

In order to regulate and secure the operations in the blockchain system, access policies are needed and created in our prototype infrastructure. Part of the core access control policies of prototype system are depicted in Figure 6. Note that, the policies are enforced in order to give access to the operations, otherwise, operations are denied. The policy R_1 allows all the users to read the resource stored in the blockchain. R_2 grants design authority the access to device records. R_3 allows design authority to create the device. Note that, two similar policies for enabling contract manufacturers to create device type and create device are omitted here. In addition, all the other participants (distributors and users) are not allowed to create a device record, but they could update the device record by using specific transactions. R_4 allows a participant to update a device with device transfer transaction if the participant is the owner of the device. R_5 allows the receiver of a device to update the ownership and quantity in the blockchain once a valid confirmation transaction is received.

D. PERFORMANCE EVALUATION

Note that, hardcore performance evaluation of Hyperledger Fabric platform is not the major objective of this paper.

```
Rule R1 {
    description: ""
    participant: "ANY"
    operation: READ
    resource: "com.chiptracking.*"
    action: ALLOW
    }

Rule R2 {
    description: ""
    participant(r): "com.chiptracking.entity"
    operation: ALL
    resource: "com.chiptracking.deviceTypes"
    condition: (r.type == "DesignAuthority")
    action: ALLOW
    }

Rule R3 {
    description: ""
    participant(r): "com.chiptracking.entity"
    operation: ALL
    resource: "com.chiptracking.device"
    condition: (r.type == "DesignAuthority")
    action: ALLOW
    }

Rule R4 {
    description: ""
    participant(r): "com.chiptracking.entity"
    operation: UPDATE
    resource(d): "com.chiptracking.device"
    transaction(t): "com.chiptracking.transfer"
    condition: (d.owner.getIdentifier() == r.getIdentifier())
    action: ALLOW
    }

Rule R5 {
    description: ""
    participant(r): "com.chiptracking.entity"
    operation: UPDATE
    resource(d): "com.chiptracking.device"
    transaction(t): "com.chiptracking.confirmation"
    condition: (d.transferTo == r.entityID )
    action: ALLOW
    }
```

Figure 6: Access Control Policies for our proposed blockchain-based framework.

In addition, a number of works have already measured the overall performance of the Hyperledger platform with detailed metrics and comprehensive analysis [65]–[67]. The scalability and reliability of Hyperledger Fabric platform have already been proved. Here, we only measure some of the specific performance metrics of the prototype system, such as the latency and throughput of transactions and queries, in order to prove the applicability of our framework.

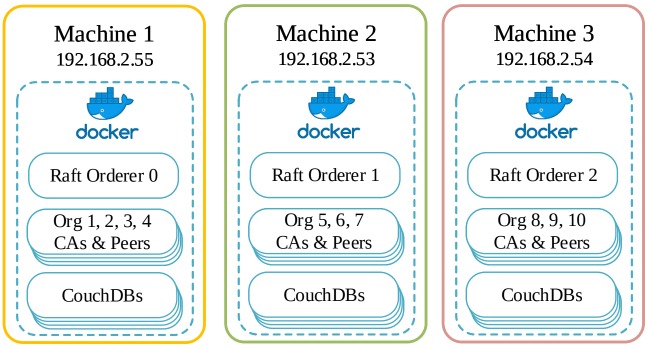


Figure 7: Proposed blockchain-based framework implemented using Hyperledger Fabric.

We setup an evaluation environment with 3 machines, each of them equipped with 8 core CPU and 16GB RAM. As depicted in Figure 7, 10 organizations in single channel with *CouchDB* state databases are created with Hyperledger Fabric 1.4.1 [68] using docker containers [69]. The block size is set to *30* with *500ms* batch timeout (details related to block size selection can be found in [67], [70]). Hyperledger Caliper [71] is used as the blockchain benchmark tool. The endorsement policy follows the default "N of N" policy , namely, a transaction needs to be endorsed by all 10 organizations. Thus, the complexity of endorsement is provided. Since Raft [72] is adopted as new consensus module in Hyperledger Fabric to replace Kafka [73], 3 RAFT orderers are deployed on these 3 machines. With these configurations, multihost blockchain system, multi-organization communication, and orderer services with fault tolerance are all provided. In addition, one client on machine 1 is created to send the transactions.

As all the *register*, *transfer* and *confirmation* transactions perform both read and write operations on blockchain, we observe similar behaviors for throughput and latency. Thus, they could be combined as read/write (R/W) transactions, and their performance mainly depends on the speed of write operations. On the other hand, *query* transactions are read only transactions, the performance should be faster than R/W transactions. Both R/W, and query transaction throughput and latency performances within different transaction rate are shown in Figure 8.

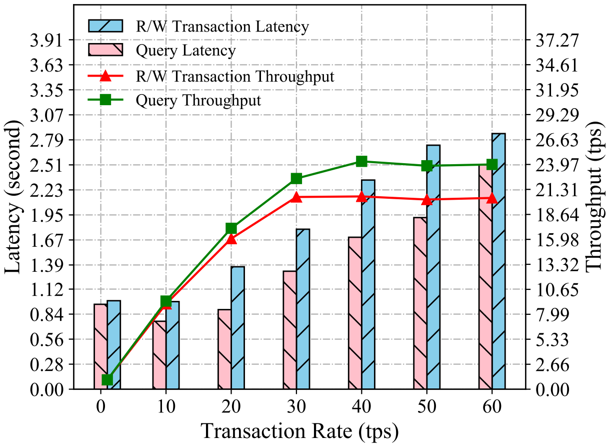


Figure 8: Latency and throughput of the proposed blockchain implementation using Hyperledger Fabric.

Generally, read only query transactions have better performance than the R/W transactions as we expected. However, R/W and query transactions reach the throughput bottleneck at 22 and 25 tps, respectively. Before the transaction rate exceeds the throughput bottleneck, transactions can be committed with a latency less than 1.5 seconds. The system is running with 500ms block timeout with a block size of 30. This signifies that a block is generated either after 30 transactions are received or the timeout of 500ms is reached. Therefore, the submitted transactions have to wait for the timeout when the transaction rate is low. For example, when the transaction rate is 1 *tps*, the average latency of query transaction is at around 0.95s (500ms timeout time and processing and transmission delay in the system).

In addition, as the transactions are continuously sent from the client, the later transactions received in each block timeout period should have lower latency. Thus, the overall average latency slightly decreases to 0.76s, when the transaction rate reaches 10 *tps*. Moreover, the later transactions need to be held at the orderers if the transaction rate exceeds the maximum throughput. The accumulated and queued transactions would have higher latency, and it can be observed that the latency keeps increasing after transaction rate exceeds 20 *tps*, which is shown in the Figure 8.

The performance is related to various factors, such as, network delay, consensus delay among multiple orderers, chaincode execution time, endorsement delay, and block validation delay. Note that, this environment is running with single channel, but the overall system throughput should always be linear to the channel numbers, as long as the system has not reached the "real" overall system bottleneck [67]. Namely, two times of throughput could be achieved by using 2 channels, 10 times could be derived with 10 channels, etc. However, a certain throughput bottleneck of the system exist, so that no matter how many channels are created, the throughput cannot exceed this threshold. In addition, one could further optimize the system performance by increasing computational power, changing endorsement policies, and using different state database, more details can be found at [67]. Note that, although replacing *CouchDB* with *LevelDB* could improve the performance by 3X, rich query is only supported in *CouchDB*.

Even with the performance in current environment, our proposed framework could still work properly. With transaction rate lower than the single channel bottleneck, both of the R/W and query transactions could be processed in 1.5s, which is sufficient for the provenance scenario. One could design the chaincode to operate multiple devices information in blockchain within one transaction. One transaction is sufficient enough to represent one shipment, as all the *IDs* of parts ready for shipping can be added in this single transaction. Note that, the objective of this paper is to demonstrate the implementation of blockchain-based framework that can address device traceability issue. As transportation of parts take longer time to reach to the receiver, a transaction can be queued and added in a appropriate time. Our blockchain based provenance framework can address the real world traceability issues.

IV. SECURITY ANALYSIS

The primary objective of the proposed blockchain-based framework is to enable traceability for electronic parts and devices. It is necessary to evaluate the security of the framework such that an adversary cannot find a way to tamper the actual traceability information. In this section, we analyze various attacks and show that the framework is resistant to such scenarios.

A. ILLEGITIMATE DEVICE REGISTRATION

Illegitimate registration occurs when an untrusted entity (*e.g.,* a rogue employee of a manufacturer) registers fake IDs into the blockchain. It can also happen if the credentials of an employee are compromised. An employee can also register a device in the blockchain unintentionally (by mistake). It is thus important for a manufacturer to identify if such registration occurs. In addition, deregistering fake devices (removing IDs) from the blockchain is possible. All the data in proposed framework is stored in chaincode or a smart contract. Even though the transactions in the blockchain are irreversible and tamper-resistant, the data stored in the smart contract or chaincode is still manageable and changeable. See the details at [74]. It is necessary to grant access these functionalities to the authorized personnel (trusted) only. Note that one can still find out when such fake IDs have been removed as the blockchain keeps track of the operations.

No matter how the fake IDs are registered in the blockchain, the fake devices must have ownership traces start from the manufacturer's warehouse, so that the provenance root could be considered as valid. Therefore, those fake IDs must be initially transferred from manufacturer to the distributor/customer in blockchain, and the distributor/customer must confirm the delivery of matched devices. Assume that an adversary has some rogue employees in manufacturer CM, which produces X amount of parts. The illegitimate registration attack consists of following phases:

1) A rogue employee at the manufacturing site uploads F number fake device IDs instead of uploading F authentic device IDs into the blockchain.
2) The manufacturer (*CM*) sends F number authentic devices to the distributors or customers. Note that *CM* is trusted in our model and produces authentic parts.
3) The adversary intercepts the shipment of the authentic parts and then replaces them with their counterfeit counterparts.
4) The distributors or customers receives the fake parts assuming they are authentic. They retrieve the device IDs and compared with the blockchain data. The verification will pass as the rogue employee at the manufacturing site uploads these IDs of the counterfeit parts.
5) Finally, the distributors or customers send the confirmation transaction in blockchain and the transactions are recorded permanently.

This entire attack is valid only when the authentic devices can be smoothly transferred out from the manufacturing site. This can easily be implemented by an additional verification stage to certify whether these devices are authorized to leave the manufacturing site. A query in the blockchain will reveal whether these devices are present in the system. If this step is followed, no authentic device will leave the site. However, if adversary launches the aforementioned attack by uploading $2F$ number of device IDs (both F number of counterfeit and authentic devices) into the blockchain (Step 1). In this case, authentic devices will pass the verification stage as their IDs are already in the system. Fortunately, this can be detected as the inventory will show additional F devices which are not present at the site. Moreover, this can also be detected during registering the fake IDs as there will be a mismatch between the actual production number and registration number in the blockchain.

B. ILLEGITIMATE TRANSFER

The illegitimate transfer occurs when a illegitimate and incorrect transfer transaction is accidentally (faulty operation) or intentionally (attack) sent. Suppose manufacturer M sends out some electronic parts to distributor D_1, however, M accidentally sends a device transfer transaction to distributor D_2. In this case, the actual devices are held by D_1, but the corresponding transaction is not available for D_1 to make the confirmation. D_1 needs to request manufacturer M to resend the transfer in the blockchain. Meanwhile, as the D_2 does not receive any devices, it is not reasonable for D_2 to confirm and pay for the devices. Thus, illegitimate transfers could not be confirmed. Similarly, suppose a manufacturer M is compromised and all its devices are transferred out by an adversary. This attack can also be detected easily since no actual physical devices are sent to that distributor, the receiver will not confirm the transfer or make the payment. There is no reason for a distributor to confirm a transaction until it receives the actual number of devices. The illegitimate transfer transactions could be cancelled or rejected. This is achieved by enabling *time-to-live* in transfer transaction, or setting a failed confirmation flag in confirmation transaction. In addition, optional functions can be created to directly cancel a illegitimate transfer before it has been confirmed, and this functionality should be accessible to authorized personnel only.

C. ILLEGITIMATE OFF-CHAIN DISTRIBUTION

The owners of the electronic parts could sell parts to the the open market, such as independent distributors or brokers, who are not a member of our proposed blockchain-based framework. In such cases, the distribution of the devices are not recorded in the blockchain. We refer this as an illegitimate off-chain distribution. One could also buy a part from an off chain market if he/she is not concerned about the authenticity of the part. In such cases, it is not recommended to sell or purchase the parts off chain as it would not allow us to guarantee the full traceability of parts. We could not enforce the authenticity for all the manufacturers', distributors', and customers' trades and therefore the record of the electronic parts in the provenance system as well, so only the devices recorded in the system from trusted suppliers are protected from counterfeiting and tampering.

V. CONCLUSION

We presented a novel blockchain based framework to provide traceability for electronic parts in the supply chain. For each device registered and distributed in the framework, one could track its origin, trace of travel, and the bill of materials in an efficient and reliable manner. All the manufacturers, distributors, and end users or customers could benefit from the framework, since it helps to protect the supply chain from counterfeit devices. We implemented our proposed framework using Hyperledger Fabric and performed detailed performance evaluation on throughput and latency. We performed a comprehensive security analysis for this framework to ensure that it is secure and reliable. Additional research is needed to explore the use of PUF and other unique device IDs, which could help to link the physical device to blockchain in tamper-resistant manner.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation under Grant Number CNS-1755733 and Intramural Grants Program (IGP) from Auburn University.

References

[1] M. M. Tehranipoor, U. Guin, and D. Forte, Counterfeit Integrated Circuits: Detection and Avoidance. Springer, 2015.

[2] IHS iSuppli, "Top 5 Most Counterfeited Parts Represent a $169 Billion Potential Challenge for Global Semiconductor Market," 2011.

[3] U. Guin, K. Huang, D. DiMase, J. Carulli, M. Tehranipoor, and Y. Makris, "Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain," Proceedings of the IEEE, pp. 1207–1228, 2014.

[4] U. Guin, D. DiMase, and M. Tehranipoor, "Counterfeit integrated circuits: Detection, avoidance, and the challenges ahead," Journal of Electronic Testing, vol. 30, no. 1, pp. 9–23, 2014.

[5] M. M. Tehranipoor, U. Guin, and S. Bhunia, "Invasion of the hardware snatchers," IEEE Spectrum, vol. 54, no. 5, pp. 36–41, 2017.

[6] B. Cyr, J. Mahmod, and U. Guin, "Low-cost and secure firmware obfuscation method for protecting electronic systems from cloning," IEEE Internet of Things Journal, vol. 6, no. 2, pp. 3700–3711, 2019.

[7] U. Guin, S. Bhunia, D. Forte, and M. M. Tehranipoor, "Sma: A system-level mutual authentication for protecting electronic hardware and firmware," IEEE Transactions on Dependable and Secure Computing, vol. 14, no. 3, pp. 265–278, 2016.

[8] J. Robertson and M. Riley, "The big hack: How china used a tiny chip to infiltrate u.s. companies," 2018.

[9] C. A. Nissen, D. J. E. Gronager, R. S. Metzger, and H. Rishikof, "Deliver uncompromised: A strategy for supply chain security and resilience in response to the changing character of war," 2018, https://www.mitre.org/sites/default/files/publications/pr-18-2417-deliver-uncompromised-MITRE-study-8AUG2018.pdf.

[10] NIST, "Framework for improving critical infrastructure cybersecurity," 2018. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf

[11] Bill Eklow, "ECID vs Device ID," 2006. [Online]. Available: btw.tttc-events.org/material/BTW10/Presentations/Session\%205.2.pptx

[12] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, "Silicon physical random functions," in Proceedings of the ACM Conference on Computer and Communications Security (CCS). ACM, 2002.

[13] G. Suh and S. Devadas, “Physical Unclonable Functions for device authentication and secret key generation,” in Proc. of ACM/IEEE on Design Automation Conference, 2007.

[14] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, “Fpga intrinsic pufs and their use for ip protection,” in International workshop on Cryptographic Hardware and Embedded Systems. Springer, 2007.

[15] W. Wang, A. Singh, U. Guin, and A. Chatterjee, “Exploiting power supply ramp rate for calibrating cell strength in SRAM PUFs,” in IEEE Latin-American Test Symposium, 2018.

[16] T. Rahman, D. Forte, J. Fahrny, and M. Tehranipoor, “Aro-puf: An aging-resistant ring oscillator puf design,” in Proceedings of the conference on Design, Automation & Test in Europe. European Design and Automation Association, 2014, p. 69.

[17] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. D. Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolic, S. W. Cocco, and J. Yellick, “Hyperledger fabric: A distributed operating system for permissioned blockchains,” 2018.

[18] G-19A Test Laboratory Standards Development Committee, “Test Methods Standard; General Requirements, Suspect/Counterfeit, Electrical, Electronic, and Electromechanical Parts,” 2016, https://saemobilus.sae.org/content/as6171.

[19] U. Guin, N. Asadizanjani, and M. Tehranipoor, “Standards for hardware security,” GetMobile: Mobile Computing and Communications, vol. 23, no. 1, pp. 5–9, 2019.

[20] U. Guin, W. Wang, C. Harper, and A. D. Singh, “Detecting recycled socs by exploiting aging induced biases in memory cells,” in 2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST). IEEE, 2019, pp. 72–80.

[21] U. Guin, D. DiMase, and M. Tehranipoor, “A comprehensive framework for counterfeit defect coverage analysis and detection assessment,” Journal of Electronic Testing, vol. 30, no. 1, pp. 25–40, 2014.

[22] X. Zhang, K. Xiao, and M. Tehranipoor, “Path-delay fingerprinting for identification of recovered ICs,” in Proc. International Symposium on Fault and Defect Tolerance in VLSI Systems, October 2012.

[23] Y. Zheng, A. Basak, and S. Bhunia, “CACI: Dynamic current analysis towards robust recycled chip identification,” in Design Automation Conference (DAC), June 2014, pp. 1–6.

[24] Y. Zheng, X. Wang, and S. Bhunia, “SACCI: Scan-based characterization through clock phase sweep for counterfeit chip detection,” IEEE Trans. Very Large Scale Integration (VLSI) Systems, 2014.

[25] H. Dogan, D. Forte, and M. Tehranipoor, “Aging analysis for recycled FPGA detection,” in IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), Oct 2014.

[26] Y. Zheng, X. Wang, and S. Bhunia, “SACCI: Scan-based characterization through clock phase sweep for counterfeit chip detection,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 23, no. 5, pp. 831–841, 2015.

[27] X. Zhang, N. Tuzzio, and M. Tehranipoor, “Identification of recovered ICs using fingerprints from a light-weight on-chip sensor,” in Proc. IEEE-ACM Design Automation Conference, June 2012.

[28] X. Zhang and M. Tehranipoor, “Design of on-chip lightweight sensors for effective detection of recycled ICs,” IEEE Transactions on Very Large Scale Integration Systems, pp. 1016–1029, 2014.

[29] M. Alam, S. Chowdhury, M. Tehranipoor, and U. Guin, “Robust, Low-Cost, and Accurate Detection of Recycled ICs using Digital Signatures,” in IEEE Int. Symposium on Hardware Oriented Security and Trust (HOST), 2018.

[30] P. Chowdhury, U. Guin, A. D. Singh, and V. D. Agrawal, “Two-pattern ΔI_{DDQ} test for recycled IC detection,” in Proc. 32nd International Conference on VLSI Design, 2019, pp. 82–87.

[31] G-19CI Continuous Improvement, “Counterfeit Electronic Parts; Avoidance, Detection, Mitigation, and Disposition,” 2009, https://saemobilus.sae.org/content/as5553.

[32] CTI, “Certification for Counterfeit Components Avoidance Program,” 2011, http://www.cti-us.com/pdf/CCAP101Certification.pdf.

[33] IDEA, “Acceptability of Electronic Components Distributed in the Open Market,” 2017, http://www.idofea.org/products/118-idea-std-1010b.

[34] K. Huang, J. Carulli, and Y. Makris, “Parametric counterfeit IC detection via Support Vector Machines,” in Proc. International Symposium on Fault and Defect Tolerance in VLSI Systems, 2012, pp. 7–12.

[35] Z. Guo, M. T. Rahman, M. M. Tehranipoor, and D. Forte, “A zero-cost approach to detect recycled soc chips using embedded sram,” in IEEE Int. Symp. on Hardware Oriented Security and Trust, 2016.

[36] T.-H. Kim, R. Persaud, and C. Kim, “Silicon odometer: An on-chip reliability monitor for measuring frequency degradation of digital circuits,” Solid-State Circuits, IEEE Journal of, vol. 43, no. 4, pp. 874–880, April 2008.

[37] U. Guin, X. Zhang, D. Forte, and M. Tehranipoor, “Low-cost on-chip structures for combating die and IC recycling,” in Proc. of ACM/IEEE Design Automation Conference, 2014.

[38] U. Guin, D. Forte, and M. Tehranipoor, “Design of accurate low-cost on-chip structures for protecting integrated circuits against recycling,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 4, pp. 1233–1246, 2016.

[39] K. He, X. Huang, and S. X. D. Tan, “Em-based on-chip aging sensor for detection and prevention of counterfeit and recycled ics,” in IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Nov 2015, pp. 146–151.

[40] M. Miller, J. Meraglia, and J. Hayward, “Traceability in the age of globalization: A proposal for a marking protocol to assure authenticity of electronic parts,” in SAE Aerospace Electronics and Avionics Systems Conference, October 2012.

[41] M. Crosby, P. Pattanayak, S. Verma, V. Kalyanaraman et al., “Blockchain technology: Beyond bitcoin,” Applied Innovation, vol. 2, no. 6-10, p. 71, 2016.

[42] M. Pilkington, “11 blockchain technology: principles and applications,” Research handbook on digital transformations, vol. 225, 2016.

[43] K. Toyoda, P. T. Mathiopoulos, I. Sasase, and T. Ohtsuki, “A novel blockchain-based product ownership management system (poms) for anti-counterfeits in the post supply chain,” IEEE Access, vol. 5, pp. 17 465–17 477, 2017.

[44] F. 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. IEEE, 2017, pp. 1–6.

[45] S. A. Abeyratne and R. P. Monfared, “Blockchain ready manufacturing supply chain using distributed ledger,” 2016.

[46] H. M. Kim and M. Laskowski, “Toward an ontology-driven blockchain design for supply-chain provenance,” Intelligent Systems in Accounting, Finance and Management, vol. 25, no. 1, pp. 18–27, 2018.

[47] F. 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). IEEE, 2016, pp. 1–6.

[48] T. Bocek, B. B. Rodrigues, T. Strasser, and B. Stiller, “Blockchains everywhere-a use-case of blockchains in the pharma supply-chain,” in 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM). IEEE, 2017, pp. 772–777.

[49] Q. Lu and X. Xu, “Adaptable blockchain-based systems: A case study for product traceability,” IEEE Software, vol. 34, no. 6, pp. 21–27, 2017.

[50] S. Chen, R. Shi, Z. Ren, J. Yan, Y. Shi, and J. Zhang, “A blockchain-based supply chain quality management framework,” in 2017 IEEE 14th International Conference on e-Business Engineering (ICEBE). IEEE, 2017, pp. 172–176.

[51] K. Biswas, V. Muthukkumarasamy, and W. L. Tan, “Blockchain based wine supply chain traceability system,” in Future Technologies Conference, 2017.

[52] K. A. Clauson, E. A. Breeden, C. Davidson, and T. K. Mackey, “Leveraging blockchain technology to enhance supply chain management in healthcare,” Blockchain in Healthcare Today, 2018.

[53] U. Guin, P. Cui, and A. Skjellum, “Ensuring proof-of-authenticity of iot edge devices using blockchain technology,” IEEE International Conference on Blockchain, 2018.

[54] X. Xu, F. Rahman, B. Shakya, A. Vassilev, D. Forte, and M. Tehranipoor, “Electronics supply chain integrity enabled by blockchain,” ACM Trans. Des. Autom. Electron. Syst., vol. 24, no. 3, pp. 31:1–31:25, May 2019.

[55] M. N. Islam and S. Kundu, “Enabling ic traceability via blockchain pegged to embedded puf,” ACM Transactions on Design Automation of Electronic Systems (TODAES), vol. 24, no. 3, p. 36, 2019.

[56] B. Tukamuhabwa, M. Stevenson, and J. Busby, “Supply chain resilience in a developing country context: a case study on the interconnectedness of threats, strategies and outcomes,” Supply Chain Management: An International Journal, vol. 22, no. 6, pp. 486–505, 2017.

[57] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,âĂİ http://bitcoin.org/bitcoin.pdf,” 2008.

[58] Anonymous. White paper: Next-generation smart contract and decentralized application platform. https://github.com/ethereum/wiki/wiki/White-Paper.

[59] Filecoin, https://filecoin.io/.

[60] Storj, https://storj.io/.

[61] Age Yeh, "Trends in the global IC design service market," DIGITIMES Research, 2012.

[62] IEEE 1149.1-2013 - IEEE Standard for Test Access Port and Boundary-Scan Architecture, https://standards.ieee.org/standard/1149_1-2013.html.

[63] L. Richardson and S. Ruby, RESTful web services. " O'Reilly Media, Inc.", 2008.

[64] Hyperledger Explorer, https://www.hyperledger.org/projects/explorer.

[65] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich et al., "Hyperledger fabric: a distributed operating system for permissioned blockchains," in Proceedings of the Thirteenth EuroSys Conference. ACM, 2018, p. 30.

[66] Q. Nasir, I. A. Qasse, M. Abu Talib, and A. B. Nassif, "Performance analysis of hyperledger fabric platforms," Security and Communication Networks, vol. 2018, 2018.

[67] P. Thakkar, S. Nathan, and B. Viswanathan, "Performance benchmarking and optimizing hyperledger fabric blockchain platform," 2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), pp. 264–276, 2018.

[68] HyperLedger Fabric, https://github.com/hyperledger/fabric#releases.

[69] D. Merkel, "Docker: lightweight linux containers for consistent development and deployment," Linux Journal, vol. 2014, no. 239, p. 2, 2014.

[70] P. Cui and U. Guin, "Countering botnet of things using blockchain- based authenticity framework," The IEEE Symposium on VLSI (ISVLSI) 2019, To Appear 2019.

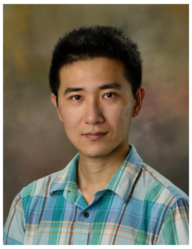
[71] Hyperledger Caliper, https://github.com/hyperledger/caliper.

[72] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in 2014 USENIX Annual Technical Conference (USENIX ATC 14), 2014, pp. 305–319.

[73] HyperLedger-Raft, https://hyperledger-fabric.readthedocs.io/en/release-1.4/whatsnew.html.

[74] Hyperledger Chaincode Functions, https://hyperledger-fabric.readthedocs.io/en/release-1.4/smartcontract/smartcontract.html.

• • •



PINCHEN CUI (S'18) received his B.E. Degree from the Department of Information Security, Beijing University of Technology, Beijing, China, in 2014 and his M.S. Degree from the Department of Computer Science and Software Engineering, Auburn University, AL, USA, in 2017. He is currently a PhD student in the Department of Computer Science and Software Engineering at Auburn University. His current research interests include Cybersecurity, Internet of Things (IoT), and Blockchain. He is a student member of both the IEEE and ACM.

JULIE DIXON will be receiving her B.S Degree in Software Engineering from the Department of Computer Science and Software Engineering, Auburn University, AL, USA, in 2019. Her specialization is in Cybersecurity, and specific interests include Risk Compliance, Hardware Security, and Digital Forensics. She participated in an internship at Caterpillar as an IT Security Specialist for the Cybersecurity Governance and Risk Management team in Peoria, IL.

UJJWAL GUIN (S'10–M'16) received his PhD degree from the Electrical and Computer Engineering Department, University of Connecticut, in 2016. He is currently an Assistant Professor in the Electrical and Computer Engineering Department of Auburn University, Auburn, AL, USA. He received his B.E. degree from the Department of Electronics and Telecommunication Engineering, Bengal Engineering and Science University, Howrah, India, in 2004 and his M.S. degree from the Department of Electrical and Computer Engineering, Temple University, Philadelphia, PA, USA, in 2010. Dr. Guin has developed several on-chip structures and techniques to improve the security, trustworthiness, and reliability of integrated circuits. His current research interests include Hardware Security & Trust, Blockchain, Supply Chain Security, Cybersecurity, and VLSI Design & Test. He is a co-author of the book *Counterfeit Integrated Circuits: Detection and Avoidance*. He has authored several journal articles and refereed conference papers. He was actively involved in developing a web-based tool, Counterfeit Defect Coverage Tool (CDC Tool), *http://www.sae.org/standardsdev/cdctool/*, to evaluate the effectiveness of different test methods used for counterfeit IC detection. SAE International has acquired this tool from the University of Connecticut. He is an active participant in SAE International G-19A Test Laboratory Standards Development Committee, and G-32 Cyber Physical Systems Security Committee. He is a member of both the IEEE and ACM.

DANIEL DIMASE is the President & CEO of Aerocyonics, Inc. and a Research Scientist at the University of Connecticut. Mr. DiMase has over 30 years of experience as an expert and recognized industry leader in Supply Chain Risk Management, Logistics, Counterfeit Parts Avoidance & Detection, Cyber Physical Systems Security, and Hardware Assurance. He is a results-driven executive with over a decade of experience in the Aerospace sector and more than three decades of experience in global sourcing, logistics and building organizations. He drives both strategic and tactical actions to advance innovation of products and services, and improvements in policies, processes, and procedures in focal areas. He has a successful track record of building teams, inspiring and motivating staff, collaborating and leading cross-functional and cross-industry groups with individuals from industry, government and academia.

Mr. DiMase has previously held leadership roles as President at SemiXchange, Inc., President at ERAI, Inc. and Director of Compliance & Quality at Honeywell Aerospace. He is the Chairman Emeritus of SAE International G-19A Test Laboratory Standards Development Committee, and Co-Chairman of the SAE G-32 Cyber Physical Systems Security Committee and Co-Chair of the SAE Distributor Process Rating Committee. He served leadership roles and provided contributions on numerous industry committees and working groups including the SAE G-19 Counterfeit Electronics Parts subcommittees, SAE G-21 Counterfeit Materiel Committee, The Aerospace Industries Association Counterfeit Parts Integrated Projects Team, and the TechAmerica Supply Chain Assurance Committee. He has been a member of the U.S. Customs and Border Protection Advisory Committee on Commercial Operations in the Intellectual Property Rights Subcommittee and the Subcommittee on Trade Enforcement and Revenue Collection, and the Government-Industry Data Exchange Program (GIDEP) Industry Advisory Group. Mr. DiMase has received the Arch T. Colwell Cooperative Engineering Medal from SAE International, the Dr. Desmond G. Newman Award for Supply Chain Excellence from the National Defense Industrial Association Manufacturing Division, and the DMSMS Achievement Special Recognition Award from the U.S. Department of Defense in recognition of superior leadership and contributions in counterfeit prevention. Mr. DiMase has an Executive MBA from Northeastern University. He has a Six-Sigma Green Belt Certificate from Bryant University. He received his Bachelor of Science degree in Electrical Engineering from The University of Rhode Island.